An Interop Publication

CONNEXIONS®

The Interoperability Report

August 1993 Volume 7, No. 8

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, 480 San Antonio Road, Mountain View, CA 94040–1219, USA.
+1 415-941-3399. Fax: +1 415-949-1779.
Toll-free: 1-800-INTEROP.
E-mail: connexions@interop.com.

ISSN 0894–5926

From the Editor

Since INTEROP 93 August is towards the end of the month, the show edition will be the *September* issue to be released on August 23. That will be our largest issue to date, a whopping 96 pages, with a special focus on the NSFNET 5 year anniversary, the Internet in general, and other conference related topics.

It seems that *cell* networking, in particular *Asynchronous Transfer Mode* (ATM), is one of the hottest topics in the industry these days. With this in mind we bring you two articles on high-speed cell networking. The first is adapted from Craig Partridge's forthcoming book *Gigabit Networking*. The article examines two cell LANs which have a ring-shaped topology, namely the *Cambridge Backbone Ring* (CBR), and the *Distributed Queue Dual Bus* (DQDB).

Our second cell article is by Randall Atkinson of the Naval Research Laboratory and examines ATM specification status, possible roles for ATM, and prospects for near-term multivendor interoperable ATM networking. ATM is truly an "emerging technology" and it will be a while before we see it widely deployed in our networks. Mark Laubach will give a talk at INTEROP entitled "ATM for Your internet—But When?" An article with the same title will appear in the September (INTEROP) issue of *ConneXions,* and I am sure that these are only the first in a series of ATM related articles.

The Internet ("TCP/IP") protocol suite has much going for it, but "plug-and-play" installation and configuration are not its strongest points. In an effort to improve this state of affairs, the Internet Engineering Task Force (IETF) has developed the *Dynamic Host Configuration Protocol* (DHCP). DHCP centralizes TCP/IP configuration, manages the allocation of IP addresses, and automates much of the configuration process. We asked J. Allard, a member of the IETF DHCP working group, to give us a tutorial on DHCP. Several vendors will be performing DHCP interoperability testing in August and we've been promised a report for an upcoming issue.

Electronic mail ("e-mail") continues to be the most widely used of all network applications. People from all walks of life have come to depend on e-mail as an essential infrastructural tool. John Klensin and Randy Bush discuss ways to expand international e-mail connectivity in a cost-effective manner.

Our final article is another brief NREN update from Washington. Mike Roberts reports on the *National Information Infrastructure Act of 1993* which is expected to be signed into law later this year. See also Mike's previous articles in the January 1993, February 1992, and June 1991 issues of *ConneXions*.

Local Area Cell-Relay Ring Networks

by Craig Partridge, BBN Systems and Technologies

Introduction

A number of prototype local area networks have been built that support *cells,* small fixed-sized packets. This article examines two cell LANs which have a ring-shaped topology.

Shared-Media Cell Networks

Local area ring networks are a type of *shared media* network. Shared-media networks are networks in which the network's physical media (fiber or copper) is accessible to multiple systems. The usual logic for shared-media networks (also called multiaccess networks) is that no single system is likely to use the entire media bandwidth for very long, so sharing the bandwidth among multiple systems is a useful cost savings. Also, the use of a shared-media tends to make support for multicasting easier, because all the systems are listening on the same wire.

The major concern in a shared-medium is making sure that when multiple systems are competing for the media's bandwidth, all the systems get an equitable fraction of the available bandwidth.

Shared-media networks can also have trouble supporting performance guarantees. If several different systems on the same media all want to reserve part of the bandwidth to support applications with performance guarantees, how are their reservations to be coordinated?

Cell networks often solve the allocation and reservation problems by establishing methods for arbitrating the right to send a cell. However, the methods for arbitrating access can vary widely and have different consequences.

Shared-Media Networks at Gigabit Speeds

Moving to gigabit speeds has an important impact on the design of shared-media networks. An important characteristic of the evolution of local area networks to gigabit speeds is that the propagation delay in a fiber is staying constant, but ever more bits can be packed into the fiber. This trend has an important implication. Access techniques, such as Ethernet's, whose performance can be influenced by the length of the network (or the number of bits that can be in the network at any one time) will have difficulty achieving high-performance, because the relative length of the network is increasing. One of the interesting features of cell LANs is how their access techniques try to achieve high performance.

Another point of difference in the various networks is their range of chosen cell sizes. At high speeds, particularly over short distances, the cell size is not an important issue. Local bandwidth is cheap enough that completely filling cells is typically not considered important, and the difference in serialization times between a 53-byte cell and a 256-byte cell is less than 2 microseconds, small enough that no system is likely to care about the difference.

Cambridge Backbone Ring

The *Cambridge Backbone Ring* (CBR) was developed as a collaborative project between the University of Cambridge and Olivetti Research. It is of interest because, the University of Cambridge arguably has more experience with cell LANs than any other organization, dating back to the first Cambridge Ring in the 1970s. Furthermore, the CBR has been in operation for some time and there is considerable experience using it to support various applications.

The CBR is a ring network in which the ring's round-trip delay is divided into a number of *frames,* which rotate around the network. Special five-bit synchronization patterns are inserted between frames to equalize timing and to fill odd spaces left if the ring delay is not an integral number of frame times.

Header	Full Monitor Type	Data 4 × 228-bit slots	Response Quality	CRC
4 bits	12 bits	1152 bits	8 bits	12 bits

Figure 1: CBR Frame Format

CBR format

The CBR frame format is shown in Figure 1. Each frame contains four *slots,* where each slot contains one cell (both data and header). In the CBR a cell is thirty-six bytes long, of which four bytes are a header containing 16-bit source and destination addresses, and thirty-two bytes (256 bits) are data.

The *header* of a CBR frame is a special 4-bit code, which is sufficiently distinct from the synchronization pattern to be recognizable even in the case of bit errors. Following the header are twelve bits of control information, three bits for each cell: a full-empty bit which indicates if the frame is filled; a type bit which indicates the data's priority and whose use in not fully defined; and a monitor bit, whose use is explained below.

The control bits are followed by four cells and then by eight more bits of control information. These eight bits contain two bits of information for each cell: the response bit, which can be used for retransmission of damaged cells; and the quality bit, which is used to indicate when corrupted cells have been detected in a frame. The control bits are followed by a 12-bit CRC which is computed on the entire frame except for the header and CRC itself.

CBR protocol

To send data over the CBR, a system looks for an empty slot by scanning the full-empty bits in each frame. When it finds an empty slot, the system inserts its data and sets the full-empty, the monitor and the quality bits. The system can also set the response and type bits if it chooses. The addresses in the cell header are set to the addresses of the sending and destination systems.

When a system recognizes its address as the destination address in the cell header, it copies the data from the slot and sets the response bit.

When a system recognizes its address as the sending address in the cell header, it must clear the full-empty bit and make the slot available for re-use. Note that it is possible for a system to fail while it has slots in use on the network. To ensure that these slots get freed, each ring has a monitoring station. When a frame passes the monitoring station, the monitoring station copies the monitor bit into the full-empty bit for each cell and clears the monitor bit. After two passes by the monitoring station, this scheme will have cleared the full-empty bit even if the sender fails to do so.

The CBR supports multicasting by allowing multiple receivers to listen for cells to a special destination address.

continued on next page

Local Area Cell-Relay Ring Networks *(continued)*

Every system on the ring computes the CRC for each frame. If the system discovers the CRC is bad, the system clears the quality bits for all the slots. In this manner, bad slots can be detected. Every time a station changes a slot, the CRC must be recomputed. Observe that by sharing the CRC over four slots, an error in one slot can corrupt all four slots. The CBR designers felt that errors would be sufficiently rare in fiber networks that it made sense to save bandwidth by using a single CRC, rather than using a CRC per slot.

The CBR designers wanted to make it possible for low cost workstations to attach to the CBR, albeit at lower bandwidths. So the CBR allows systems to selectively receive only from one slot position in the frame. Each system maintains a table mapping destination addresses to slot positions. To send, a system looks up the slot position being read by the destination and inserts the cell into a free slot in that position. Observe that for this system to work, each system must be able to send in any slot. (There is a chance in this scheme that all the low speed stations will end up using the same slot position to receive in, and waste $^3/_4$'s of the ring's bandwidth. The initial CBR design does not fix this problem, but an enhanced CBR has been designed which tries to mix traffic evenly among slots.)

Ring networks have historically prevented systems from hogging all the network bandwidth by permitting systems to only have one packet (or cell) in transit at any given time. However, relatively long propagation delay in gigabit networks and the desire to allow systems to use large portions of the bandwidth, required the CBR designers to permit a system to have multiple slots in use at one time.

Super-Tokens

Unfortunately permitting a station to use multiple slots can cause *super-token* behavior.

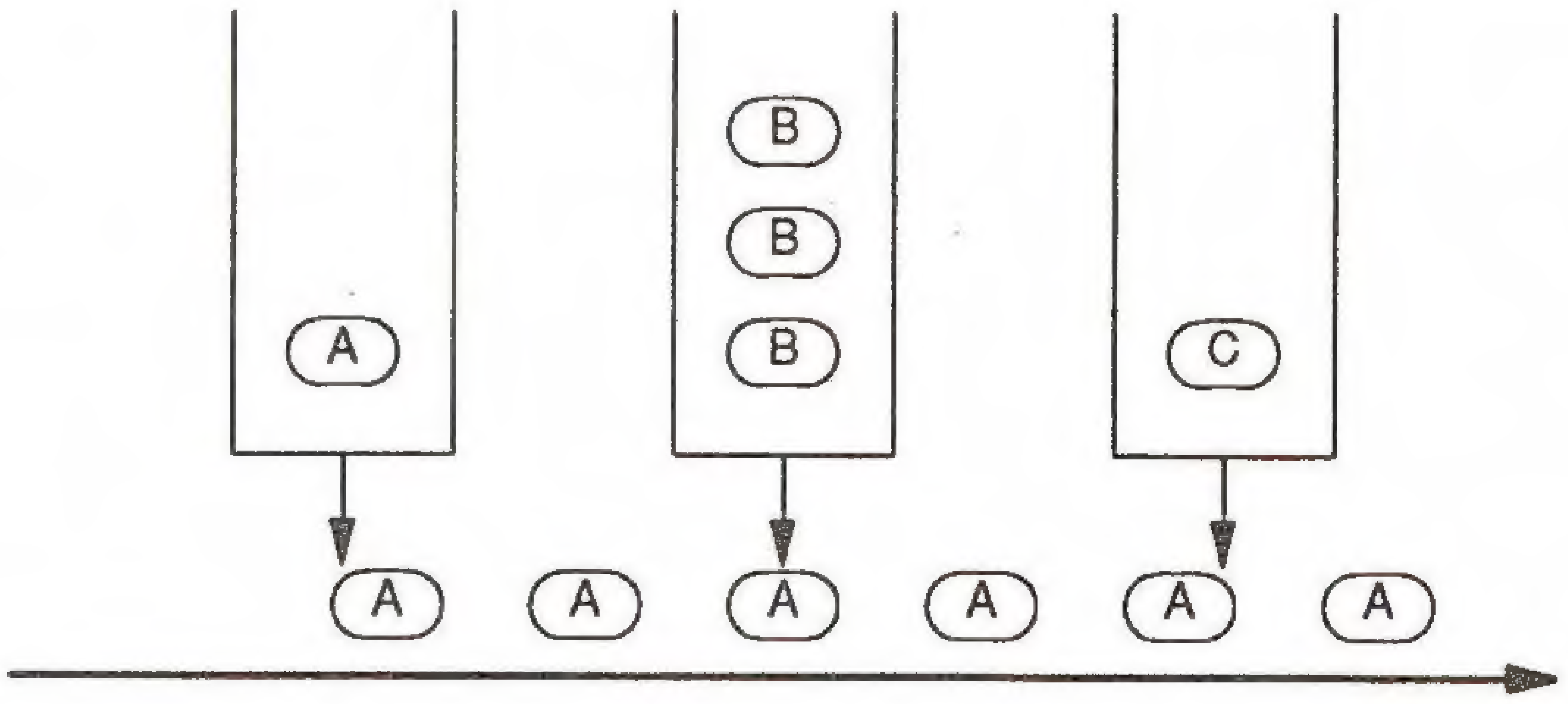

Figure 2: Super-Token Behavior (System A sending)

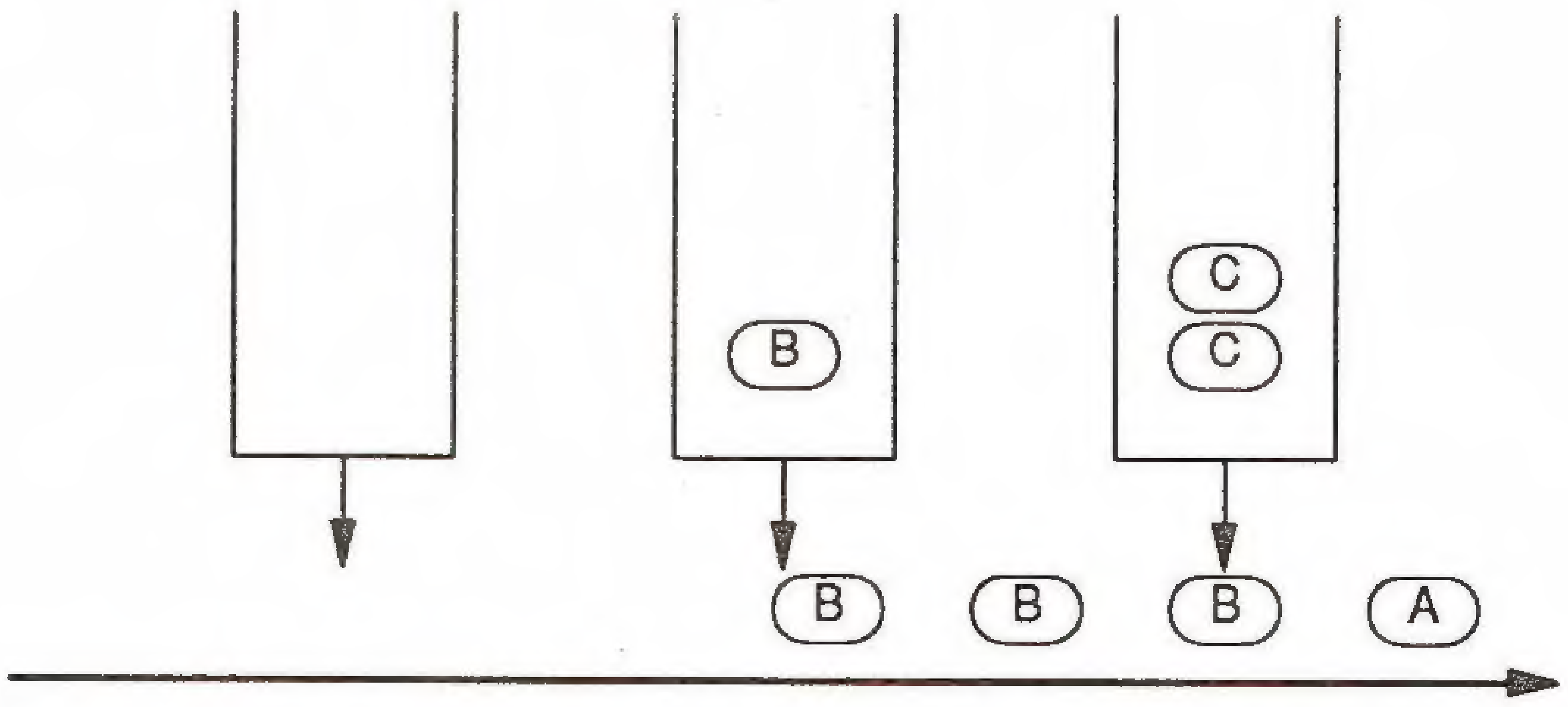

Figure 3: Super-Token Behavior (System A finished sending)

Figures 2 and 3 illustrate the concept of a super-token. System A is sending a burst of cells onto an otherwise inactive network. Figure 2 shows that while system A is sending, the shared network is in use and systems B and C are unable to send. As a result, data begins to build up in the queues at systems B and C. Finally, system A empties its queue, and now system B is able to send (Figure 3). But system C is now blocked by system B's traffic and its queue will continue to build up until system B finishes. The result is that each system gets access to the network in big clumps, rather than on a per-cell basis. The network acts very much like one that uses a single token to control which hosts get to send. Thus the term "super-token."

Why Super-Tokens?

It may not be immediately obvious why the super-token behavior persists. Looking at Figure 3 one might reasonably wonder what happens if system A sends a cell. Wouldn't that cell get mixed into system B's cells, and thus, over time, could we not expect that the token would get broken up? It turns out that there are two reasons that tokens tend to persist.

The first reason has to do with how cells get cleared from the network. Recall that the CBR is a ring, so all cells which are sent out the right side of the figures are looping around the ring and coming back in from the left. These cells' slots are considered full until the sender clears the full bit. So if system A sends long enough, all the cell slots on the network will contain system A's data, and will appear full to any system except system A, which is clearing the slots. System B only gets to send once system A stops refilling its cleared slots with more cells. Once system A stops sending, unless it gets a new cell to send very quickly, it will pass on all its free cells to system B, which then monopolizes the network in its turn.

The second reason tokens persist has to do with how higher layer protocols tend to work. To keep senders from sending data faster than receivers can accept it, higher layer protocols typically use some form of flow control to limit how much data a sender can transmit before receiving an acknowledgement. This observation implies that the reason system A stopped sending data in Figure 3 is that all of its higher layer protocols have sent as much data as they can without an acknowledgement. (No other system could acknowledge the data received because system A was hogging all the cell slots). Until the systems to which system A was sending get access to the network, system A's higher protocols will be waiting for acknowledgements and will not have more data to send. Thus higher layer flow control tends to cause super-token behavior because the only way a system can get more data to transmit is to give up the super-token so that its receivers can send back acknowledgements. (There are some protocols that do not require acknowledgements. But they are relatively rare. It is also worthy of mention that flow control can cause super-token behavior in point-to-point networks. Super-token behavior was first observed by Van Jacobson as a phenomenon on the ARPANET.)

Super-Token avoidance

To avoid super-token behavior, the CBR does not permit stations to fill successive slots. Also, when sending systems free a slot they have just used, they may not refill the slot but must pass it on.

Observe too, that by permitting a station to use multiple slots and optionally supporting retransmission of cells, the CBR may cause retransmitted cells to arrive out of order at the receiver. This behavior violates ATM's expectations but the CBR predates ATM. Furthermore, there's no requirement (except perhaps market pressure) that a cell network conform to ATM.

continued on next page

Local Area Cell-Relay Ring Networks *(continued)*

The CBR has limited support for performance guarantees. Each station may transmit on all the slots in the ring, once every *n* ring rotation times, so the maximum delay is roughly bounded. There is no way to allocate a particular share of the ring bandwidth. However, just the simple delay constraints are apparently sufficient to allow experimentation with multimedia applications.

The CBR is a good example of a cell ring network. It illustrates the basic problems (such super token behavior and the need for hosts to use multiple slots per ring revolution) that any ring designer must confront.

One feature that the CBR does not have is slot removal, in which the receiver frees the slot or an erasure station detects and free slots that have been read. Freeing slots permits other nodes to reuse the slot as the slot goes around the ring between the receiver and the sender. Since, on average, a cell only travels half way around the ring to get from its sender to its receiver, having the receiver free the slot will typically double the effective bandwidth of the ring.

IEEE 802.6 (DQDB)

Distributed Queue Dual Bus (DQDB) is a joint standard of the Institute of Electrical and Electronic Engineers (IEEE) and the American National Standards Institute (ANSI). It is usually referred to either by its acronym (DQDB) or its IEEE standard number (802.6). It is derived from an earlier technology called *QPSX*. DQDB is of interest largely because its standard committee worked carefully with the ATM standards bodies to try to ensure that DQDB local networks would interconnect with ATM long-distance networks. DQDB's cell and header sizes are the same as that of ATM and the header formats are almost identical and AAL 3/4 and the DQDB MAC protocol are identical.

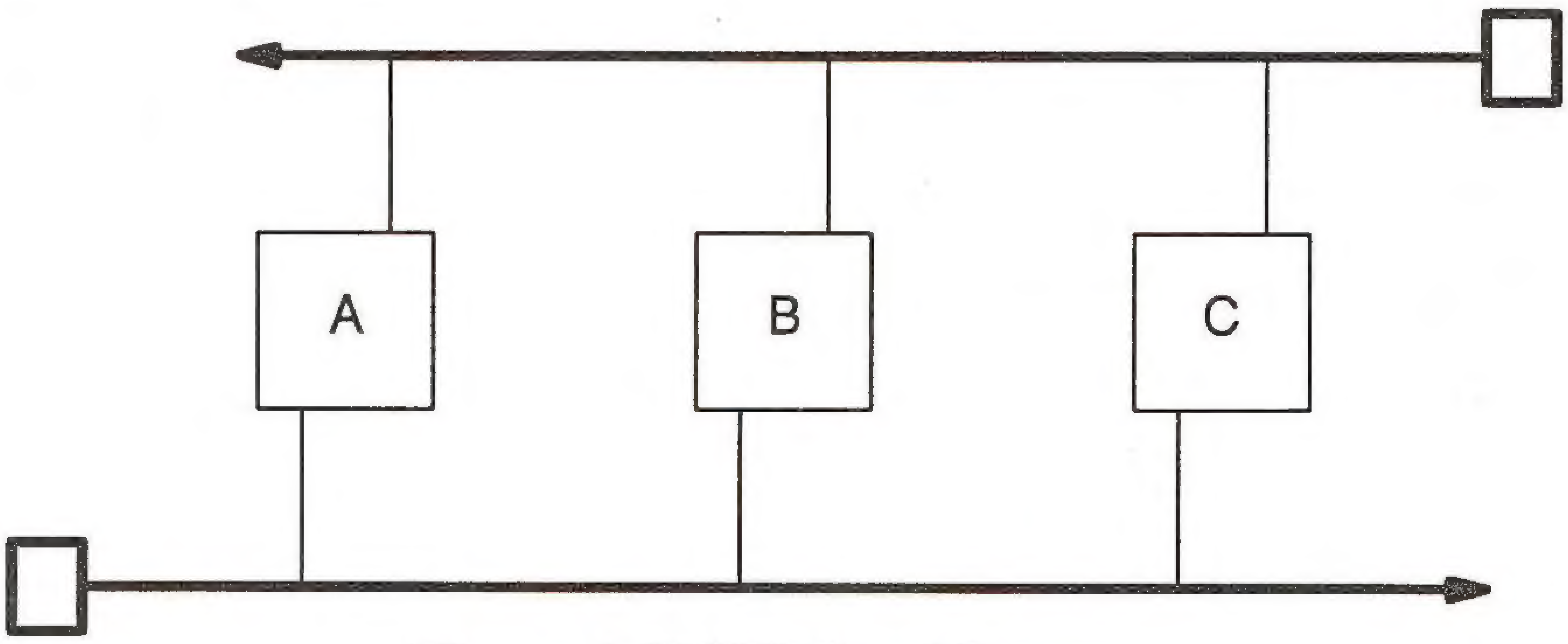

Figure 4: DQDB Dual Busses

In a DQDB network, every system is connected to two unidirectional busses, as shown in Figure 4. The two busses transmit in opposite directions, so there exists a transmission path from every system to every other system. The busses are slotted and each slot can hold one 48-octet cell plus a 5-octet cell header. The slots are generated by devices at the head of each bus and transmitted "downstream."

The dual bus architecture is typically implemented over a physical ring network, as shown on the left of Figure 5 The advantage of this approach is that if a link between two nodes fails, then the ring can reconfigure itself to be a true bus, where the two end nodes are the nodes on either side of the failure (as shown on the right of Figure 5).

Like the CBR, DQDB places cells in individual slots. However, unlike the CBR, slots are not grouped into frames. In addition, to support traffic with real-time delivery requirements, DQDB allows slots to be permanently reserved (or according to the language of the DQDB standard, "pre-arbitrated").

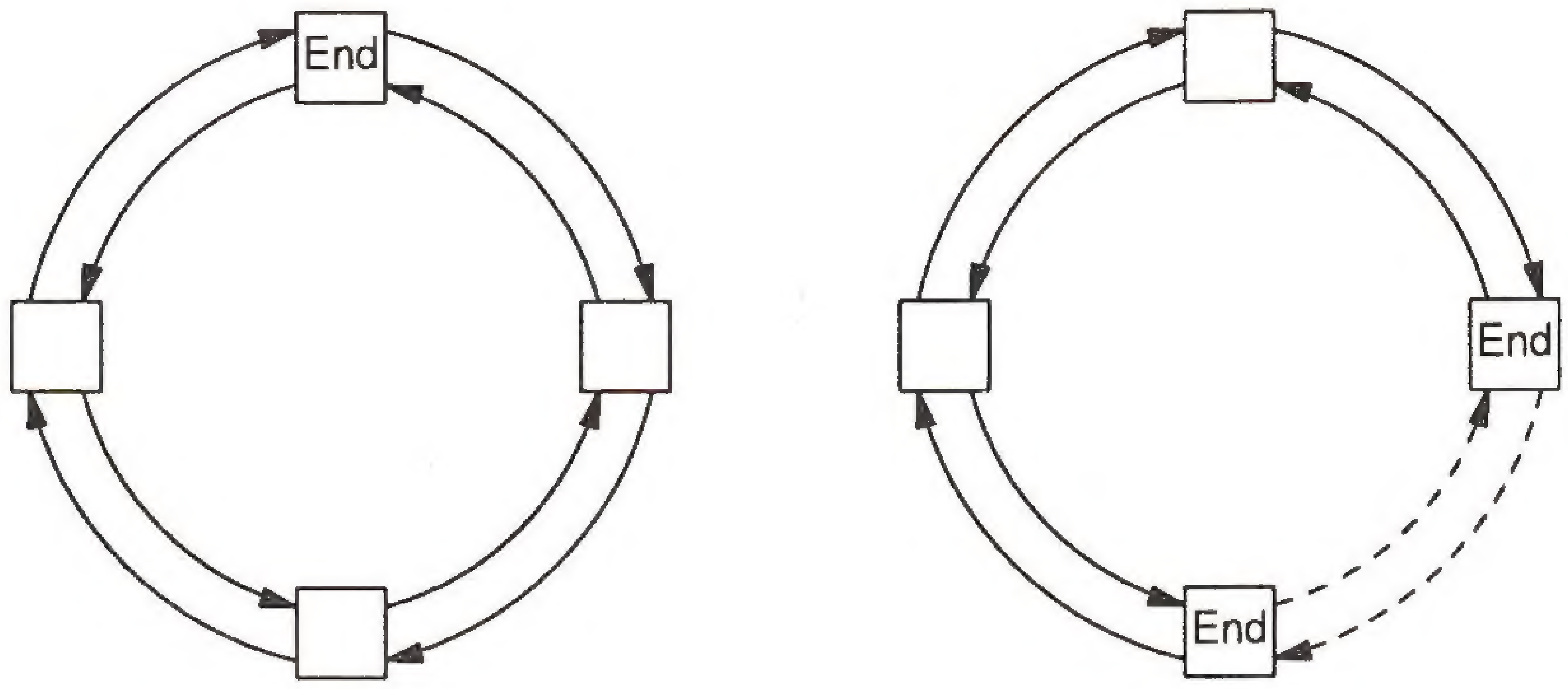

Figure 5: DQDB Implemented over a Physical Ring

DQDB format

The DQDB slot format is shown in Figure 6. The fields are identical to ATM format at the *User-Network Interface* (UNI), with three important exceptions:

- The 4-bit *Generic Flow Control* (GFC) field has been replaced by an 8-bit *Access Control Field* (ACF), which is used to control access to the DQDB busses.
- Instead of a VPI and VCI, DQDB just has a VCI.
- The control bit fields are slightly different. DQDB has a 2-bit priority field and a 2-bit payload field, while the ATM UNI has a 3-bit payload field and a 1-bit cell loss priority field. Originally, these control bit fields were the same, but the change to ATM to accommodate the user signaling bit in the payload field caused the two standards to diverge. (The DQDB standard had been issued before ATM was changed).

DQDB protocol

Access to slots is managed using the ACF and the VCI. There are two different modes of operation, one for regular traffic, the other for reserved (real-time) traffic.

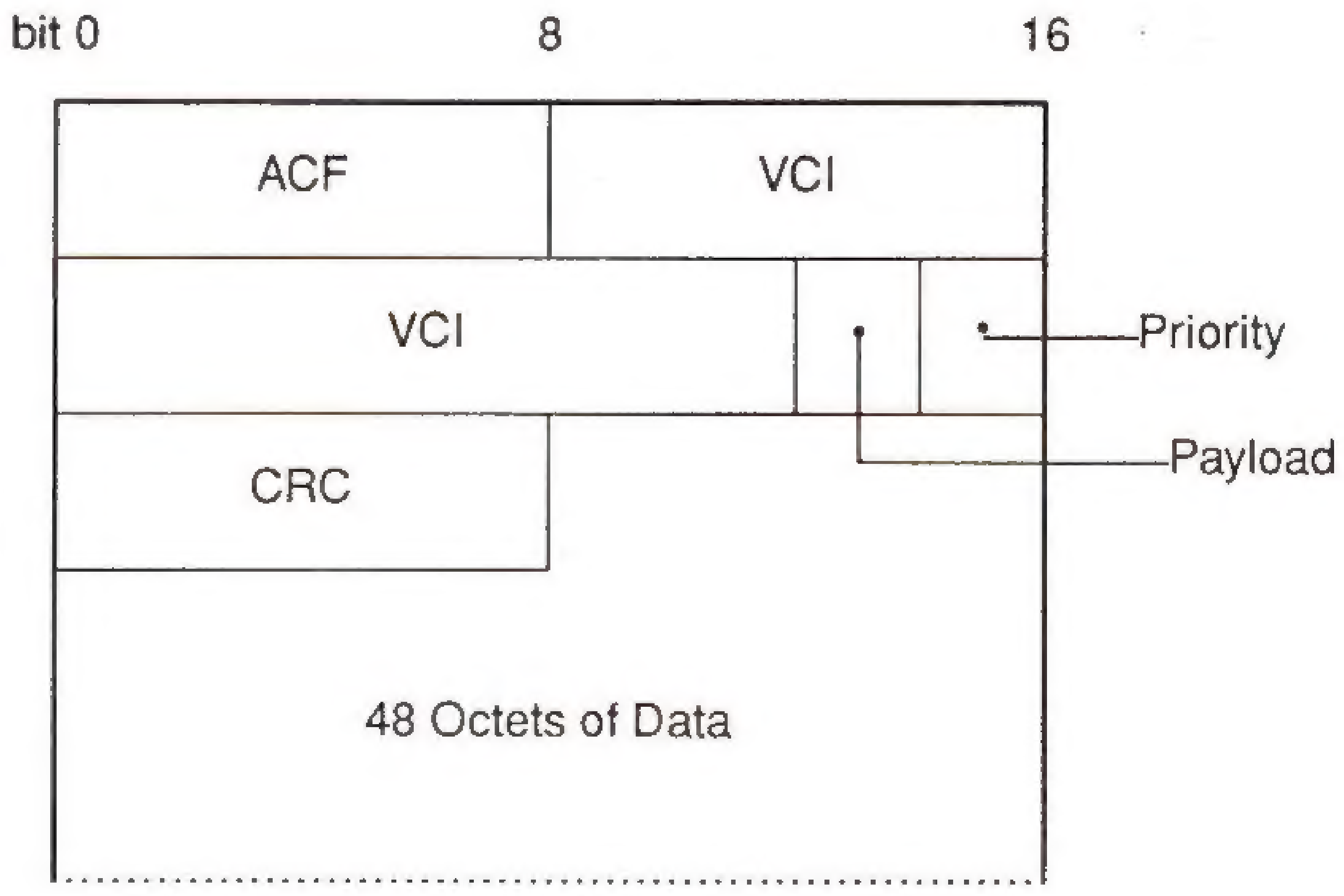

Figure 6: DQDB Slot Format

For regular traffic, DQDB uses a distributed arbitration scheme. To send a cell, a system must reserve a slot on the appropriate bus (i.e., the bus on which the sending system is "upstream" from the destination).

continued on next page

Local Area Cell-Relay Ring Networks *(continued)*

To reserve a slot on one bus (call it the "sending bus" for this cell), a system examines cells on the *other* bus (the "reservation bus"), looking for a slot in which a request bit in the ACF is 0, and sets that bit to 1. DQDB supports three levels of priorities, so there are three possible bits that can be set, depending on the priority the system assigns to the cell it wants to send. The request bit indicates to systems upstream of the sender on the sending bus that a slot has been requested at a given level of priority. The sending system then waits for its free slot on the sending bus, and when the free slot comes, fills the slot with its cell, and sets the VCI.

Free and empty slots are distinguished by two bits of the ACF. (Two bits are required to distinguish full slots from empty slots because there are two types of slots: regular slots and pre-arbitrated slots. Thus the network has to distinguish between full and empty regular slots and pre-arbitrated slots, which requires three distinct values.)

The sender employs counters to determine which free slot it should use. The sender keeps a count, for each priority level, of the requests for slots that it has seen on the reservation bus. Whenever the node sees a request bit of the same or higher priority level turned on, it increments the counter. Whenever the node sees a free slot of the same or higher priority level on the sending bus, it decrements the counter. When a node requests a free slot, it copies the request counter into a count-down counter. This counter counts down as it sees free slots on the sending bus until its counter is zero. The system then places its cell in the next slot. If higher priority requests are made on the reservation, while the count-down counter is being used, then the count-down counter must be incremented, to let the higher-priority traffic through first. Keep in mind that since both buses may be used for sending and for reservations, two sets of counters must be maintained. Note that this queueing scheme is distributed—the management of the slots is done separately at each system, by observing traffic and requests on each bus.

A matter of fairness

Unfortunately, it turns out that the DQDB allocation scheme does not fairly allocate slots. If two systems are sending at the same time, it is extremely unlikely that the systems will share the bandwidth equally. To see why this might be so, look back at Figure 4 and consider a situation in which systems A and B are both transmitting long (multiple cell) messages to C. Assume the network was initially idle and that system A started sending its message first. In this situation, system A's requests for free slots are immediately served because its counter of previous requests is always 0. Then system B starts sending. To send, system B makes a request on the top bus. Until system A hears system B's request, system A will keep filling slots, and when system A hears system B's request, system A will let just one free cell through, and then resume filling slots. Only when system B finally sees the free slot can it send its cell and request another free slot. Thus the amount of bandwidth that system B receives depends upon its physical distance from system A. Finding a perfect solution to this problem is rather hard, so DQDB uses a heuristic. Every so often, a system is required to let a free slot that the system had reserved go past. Forcing systems to occasionally skip using a free slot means that a system that is accidentally hogging the bus (like System A in the example) will slowly yield its bandwidth to other systems.

Another problem with the DQDB scheme is that, at high load, the time to get the right to send a cell depends on the length of the bus. Because the time to serialize a cell is much shorter than the time it takes for a bit to travel the length of the bus, the result is that the time required to send a cell is heavily influenced by the bus length.

DQDB handles real-time or pre-arbitrated traffic differently. Each stream of pre-arbitrated traffic is given a unique VCI. Reserved slots are generated by the head of the appropriate bus with their VCI fields set to the VCI of the channel using each slot. A bit in the ACF is set to indicate this slot has been preassigned. The sending node watches for preassigned slots with the right VCI and puts its cells into those slots. Note that while the mechanism for using pre-arbitrated cells has been defined, the DQDB standard currently does not define how to set up VCIs. So currently, this service is useless.

DQDB addressing

Addressing in DQDB is a little strange. Pre-arbitrated slots use the VCI field. But this means there's no space in the header for addressing regular slots. So DQDB defines one VCI to be special (the VCI of all ones) and requires each system to reassemble all the cells sent on the special VCI back into packets and then examine the destination address in each packet to see which system the packet is destined for. Packets are fragmented into cells using the segmentation and convergence layers of AAL 3/4 (using a distributed MID assignment algorithm, to ensure no two systems use the same MID). The data contents of the AAL 3/4 convergence layer data unit is a packet in IEEE 802 format. The DQDB standard permits systems to use one of six different address formats in the 802 header; most of these schemes support multicast addressing.

The result of this addressing and adaptation layer cake is that DQDB can simultaneously support two types of traffic: reserved traffic using cells and regular IEEE 802 style packets. This dual type of service has considerable appeal. However, the down-side is that DQDB is hard-wired to use AAL 3/4 (with its high overhead), while the data communications community largely favors AAL 5 (SEAL). (The 802.6 standards committee is considering revising DQDB to support AAL 5 but this change will presumably take some time to effect.)

Furthermore, while the DQDB cell service was originally intended to be completely compatible with ATM, the standards have now diverged slightly. Originally it had been thought that because DQDB would seamlessly interconnect with ATM wide area networks and provide transparent support for the ATM data-oriented AAL, DQDB had an inside track to become the local area network of the late 1990s. Now, it appears that DQDB picked the wrong AAL, and may not interconnect quite so perfectly with ATM. Its future is less clear.

For further reading

[1] Greaves, D. J., Lioupis, D. & Hopper, A., "The Cambridge Backbone Ring," Proceedings of IEEE INFOCOM '90, June 1990.

[2] IEEE "Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)," July 1991.

[3] *ConneXions,* Volume 6, No. 4, April 1992, "Special Issue: Emerging Broadband Networks."

Adapted from *Gigabit Networking* by Craig Partridge. Copyright © 1994 by Addison-Wesley Publishing Company. First Printing October 1993. Reprinted by permission of Addison-Wesley, One Jacob Way, Reading, MA 01867. Call 1-800-238-9682 for more information on this book.

CRAIG PARTRIDGE holds an A.B., M.Sc. and a Ph.D from Harvard University. Since 1983 he has worked for Bolt Beranek and Newman on a variety of networking related projects including CSNET, the NSF Network Service Center (NNSC), and various projects concerned with distributed systems, IP transport protocols, and gigabit networking. He is a member of the Internet End-To-End Research Group, the Internet Engineering Task Force and a Senior Member of the IEEE. He is the past editor of ACM *Computer Communication Review* and the current editor of IEEE *Network Magazine.* E-mail: `craig@aland.bbn.com`

Towards Real ATM Interoperability

by Randall Atkinson, Naval Research Laboratory

Introduction

In April 1992, *ConneXions* published several articles on Broadband ISDN, including both *Asynchronous Transfer Mode* (ATM) networking and the *Synchronous Optical Network* (SONET) technology that lies beneath ATM. Since then, there has been much publicity about ATM networking. This article attempts to discuss ATM specification status, possible roles for ATM, and prospects for near-term multivendor interoperable ATM networking.

A review of ATM and SONET

ATM is the layer 2 and layer 3 technology in Broadband ISDN. It is the second attempt by telephone service providers to become a dominant player in the data communication business. The first attempt was ISDN. ATM makes several technological advances, including the use of optical cabling and fast cell switching. The lower portion of ATM is based on cells which contain a 5 octet header and a 48 octet payload. The upper portion of ATM consists of several different ATM Adaptation Layers, which permit upper layer protocols, such as IP or CLNP, to be encapsulated for transmission through the ATM service. ATM will provide a connection-oriented service at data rates from 51 Mbps to over 1 Gbps. ATM was originally designed to support voice calls which have a fixed data rate and well understood load characteristics. In telephony networks, ATM cells travel from source to destination through ATM switches connected with high bandwidth point-to-point SONET trunks. SONET provides a high bandwidth replacement for T1 and T3 point-to-point circuits, along with an enhanced multiplexing capability.

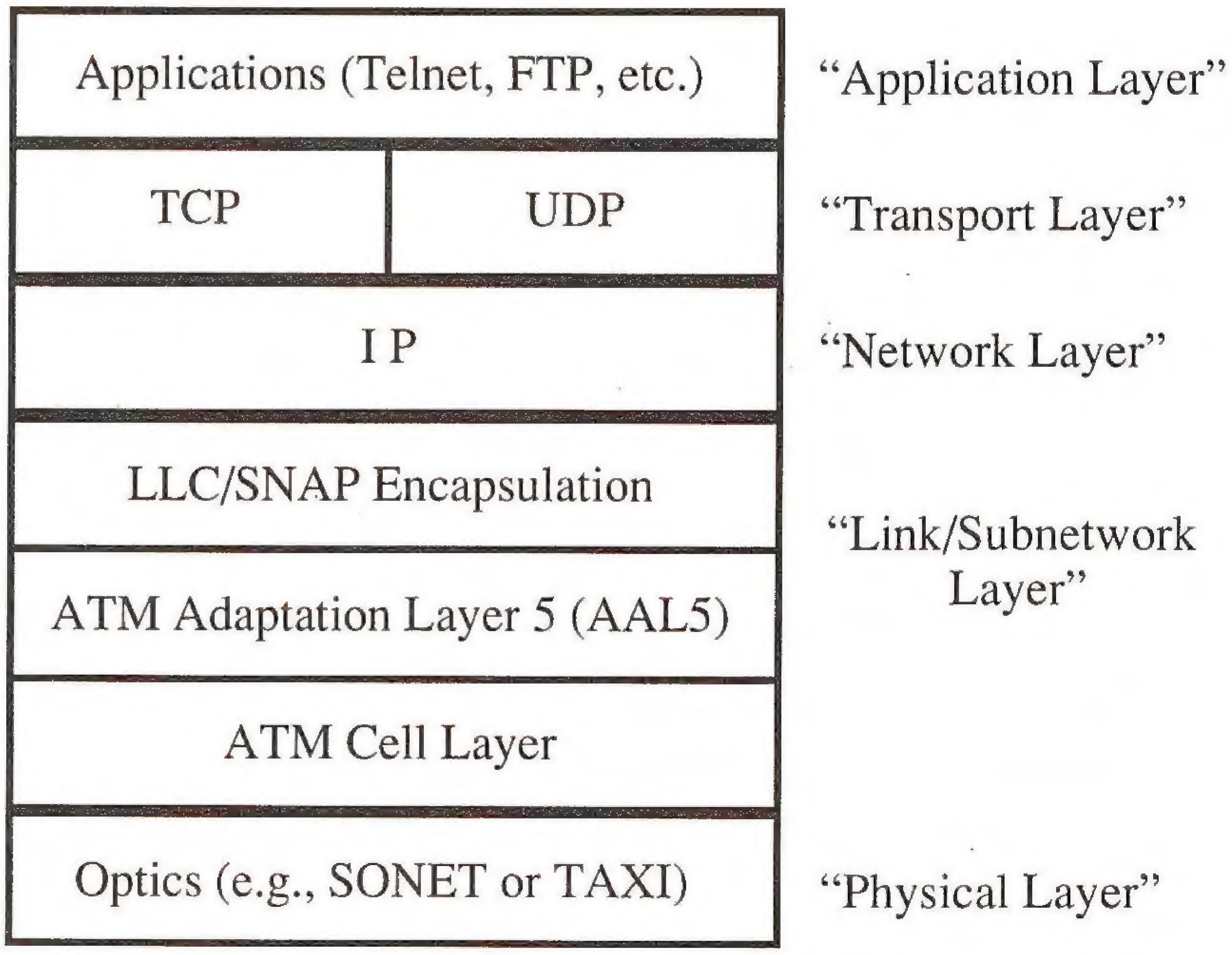

Figure 1: Example of Protocol Architecture with ATM

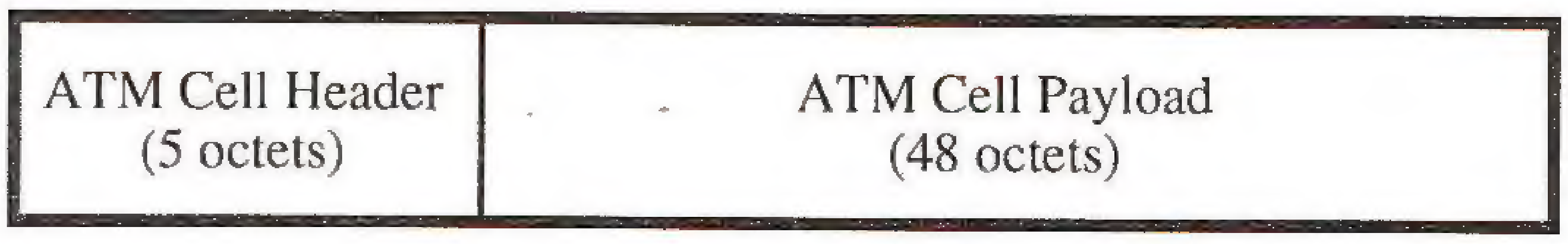

Figure 2: Basic ATM Cell Structure

Because ATM provides a connection-oriented service, it has signaling protocols that provide for call setup, call management, and call removal. For the ATM *User to Network Interface* (UNI), a derivative of the International Telecommunications Union's Q.931 signaling protocol is being developed. This derivative, currently in draft form, is widely referred to as Q.93B. Telephone service providers plan to use an enhanced version of ISDN's *Signaling System 7* protocol to provide the switch-to-switch signaling within the network. However, some vendors have suggested it would be better to extend Q.93B and use it within the network as the switch-to-switch signaling protocol. It is not clear which approach will ultimately be successful. The signaling protocols are critical components in ATM. For example, the round-trip delays in wide area networks are very large relative to the transmission speeds of ATM. Thus it is important to minimize the number of round trips required to set up an ATM circuit. Also, the addressing information in ATM is handled by the signaling protocols. ATM Cells do not contain source and destination addresses. The routing function is closely connected with the signaling protocol. A route is selected during the call setup process of the signaling protocols. Once an ATM circuit has been established, all traffic over that circuit will travel the same route. This is unlike IP or ISO CLNP, both of which use dynamic routing.

Progress towards ATM standards

The International Telecommunications Union's standards process for ATM and SONET is not moving quickly. Until the ITU's specifications are completed, true multivendor interoperability cannot exist. Many vendors have become frustrated with the slow rate of specification by the ITU. As a result, several telephone firms and computer firms have come together and formed an industry consortium, the ATM Forum, to develop common specifications more quickly. The ATM Forum uses existing ITU specifications and drafts as the basis for much of its work, but is not waiting for ITU standards process completion before issuing its own specifications.

UNI

In June 1992, the ATM Forum released its "User Network Interface (UNI), Version 2.0" specification. [1] In August 1993, the ATM Forum intends to release the Version 3.0 of that specification. This document should be widely available in September 1993. The third version of the UNI should significantly improve the interoperability between computers and switches because it includes the first phase of the ATM Forum's Q.93B-derived signaling protocol specification. However, that specification will not include all of the components needed to support full switch-to-switch interoperability.

The UNI specification includes a minimal user to network signaling protocol specification that is aligned with Q.93B. It also includes a basic specification for traffic management and several physical layer specifications. Multiple physical layer specifications are needed because SONET hardware is not yet mass produced and hence is still expensive. Multiple physical layers are also needed because customers need support for installed cabling systems. The physical layer specifications include not only SONET but also physical layers based on FDDI optics, Fiber Channel optics, T3 leased lines, and twisted-pair copper wiring. Some SONET optics and chipsets are now available but are much more expensive than FDDI chipsets owing to the fact that SONET optics are designed for long haul circuits while FDDI is designed for the local area.

continued on next page

Towards Real ATM Interoperability *(continued)*

The ATM Forum is expected to begin work in July 1993 on the *Private Network-to-Network Interface* (Private NNI). This will include specification of the signaling protocol and other components necessary for ATM switches to communicate with each other. This switch-to-switch interface needs to be scalable from LANs to WANs and must not impose an undue processing burden on the ATM switches. Without a common NNI specification, it is not possible for a fully interoperable multi-vendor ATM network to exist. The NNI will also add enhanced traffic management capabilities and support for ATM routing.

Traffic management is essential to the success of ATM in order to minimize problems with switch congestion. Switch congestion appears to be one of the challenging technical areas in ATM networking. Bursty data might cause ATM cells to arrive at an ATM switch faster than the switch can handle them. If one uses call admission control, one can preclude this, but may then inefficiently use the network bandwidth. If a switch sees more cells than it can handle, it will drop cells. If the switch doesn't select which cells to drop in a reasonable manner and if there is an adverse interaction with the upper layer retransmission algorithm, the ATM network might suffer congestion collapse. This type of collapse might be avoidable with careful design of the upper layer algorithm retransmission scheme. The issues relating to ATM switch congestion clearly need further investigation. It would be unfortunate if traffic management were fully specified before the research community has more opportunity to find a solid technical solution to this difficult problem.

Addressing

Additionally, addressing and routing are other critical standards issues. There are two kinds of addressing defined in the ATM Forum specifications. For non-telephone networks, ATM addresses have the same syntax as GOSIP NSAPs. However, these ATM addresses are not necessarily interchangeable with ISO *Connectionless Network Protocol* (CLNP) addresses. In public ATM networks operated by telephony firms, the ITU's E.164 international telephone number format may also be used. Both of the address formats provide hierarchy and are large enough to scale to very large networks. The address contains a network-dependent portion and a host-dependent portion. The host and switch exchange these two parts as part of a host registration protocol. At the completion of that exchange, both switch and host know the complete ATM address of the host. While agreeing on address syntax is very important, the more difficult problem is specifying ATM routing protocols. At present there are no standard ATM routing protocols. This work will likely be undertaken within the ATM Forum's Private NNI work that is expected to begin this July. There have been discussions within the IETF on *Virtual Circuit Short-Cut Routing* and how it might be used in a large public data network such as ATM. The ATM Forum's Private NNI working group will itself be specifying ATM routing mechanisms. The Private NNI group plans to consider information from the IETF working group as part of that effort.

ATM Adaptation Layers

The ITU originally created 4 ATM *Adaptation Layers* (AALs) for various kinds of traffic, ranging from telephony voice to computer data. The AALs provide error detection and framing for user data or upper-layer protocol data, they do not provide any error-correction or retransmission services. Error correction and data retransmission must be provided by the upper layer protocols (e.g., TCP or TP4) for users that require those services.

AAL 1 is for voice and AAL 2 is for variable-rate video. AAL 3 was designed for connection-oriented data and AAL 4 was designed for connection-less data. AAL 4 has since been dropped and AAL 3 has been renamed AAL 3/4. AAL 3/4 has its own headers inside the payload area of a cell carrying AAL 3/4 traffic. After examination of AAL 3/4, a group of computer vendors created a new, simplified AAL intended for computer networks, AAL 5. AAL 5 is designed to reduce the implementation complexity and processing intensity of AAL 3/4. Virtually all computer vendors are implementing AAL 5 instead of AAL 3/4.

IP over ATM

The *Internet Engineering Task Force* (IETF) has been working on IP over ATM for over a year now, with a focus on IP over ATM AAL5. A standards-track RFC specifying a multiprotocol encapsulation of upper layer traffic, such as IP or ISO CLNP, over ATM Adaptation Layer 5 was issued in July 1993. [2] The LLC/SNAP encapsulation method specified in that RFC must be implemented on all IP systems and is the default encapsulation method. However, other encapsulations may be used if both parties agree to the other encapsulation in advance. A consensus on using 9180 octets as the default *Maximum Transmission Unit* (MTU) for IP over AAL5 emerged within the IETF at its Amsterdam meeting and the current Internet Draft will probably become a standards-track RFC in the near term. [3] While several proposals for IP to ATM address resolution have been made over the past few months, agreement was reached in Amsterdam to use a single ARP server per IP subnetwork. Multiple implementations of the multiprotocol encapsulation for use over AAL 5 are underway and informal interoperability testing of those implementations should begin later this year. There is another internet draft called "Classical IP and ARP over ATM" which is the IETF's first complete draft specification for implementing IP and ARP over ATM AAL5. [4] This last is currently work in progress.

ATM deployment

Once the technology is fully specified, one must consider how best to use the technology. There are two commonly discussed ways of deploying ATM networks. The first is as a wide area backbone, probably with network service purchased from telephone service providers. The second use is as a LAN or campus-wide MAN. Wide-area use of ATM is clearly going to happen in the near term. As T3-based networks experience overloading, users and network providers will be forced to migrate from T3 services to ATM services in order to get more bandwidth. Initial WAN configurations appear likely to have high-performance routers connected via ATM. Later WAN configurations might also involve connecting ATM-based LANs or MANs together. ATM is likely to be the basis of most long-haul data networks by the end of this decade and will continue to be important into the next century. Firms such as Sprint and Telecom Finland have been very aggressive in developing and deploying ATM technology and in using ATM to pursue their potential data communications customers.

Similarly, several major workstation vendors have been examining the use of ATM technology for high speed LANs in high-performance workstations. There are already several vendors who are selling local-area ATM switches and workstation interface cards. While these initial products are not fully interoperable, they will become interoperable once the specifications are published. One of the driving forces behind ATM in the LAN environment is the need to connect high-performance graphics workstations with high-performance supercomputers and parallel computers.

continued on next page

Towards Real ATM Interoperability *(continued)*

Proponents of ATM LANs believe that the costs of ATM will be driven down rapidly by the combined volume of WAN and LAN sales. If costs do decline rapidly, then it might begin to make economic sense to use ATM instead of some other LAN technologies. However, the rate of installation of FDDI and Ethernet does not appear to be declining or leveling off. Not all sites have high performance computing activities and so not all sites believe that they need to have ATM today. Many sites are likely to wait a few years before deploying ATM so that the technology can be fully specified and better understood.

Lower bandwidth ATM

There is active discussion of how to send ATM over twisted-pair wiring such as is found in existing building telephone wiring. Different vendors proposals would support either ATM over shielded twisted-pair wiring or ATM over unshielded twisted-pair wiring. Support for twisted pair is important because many buildings already have such wiring in place and the cost of installing new fiber in existing buildings can be prohibitive. By reusing existing wiring to connect computers to a building-wide ATM hub or switch, ATM can be extended to the desktop at much lower cost. These twisted-pair proposals generally are lower speed than the ATM over fiber-optic wiring specifications because copper wiring has lower bandwidth and a significantly higher problem with electrical noise.

There is analogous work going on to adapt ATM for use over existing long-haul telephone trunks such as T1 and T3 circuits. ATM over T3 is clearly practical and useful. ATM over T1 is possible, however the high overhead of ATM and the relatively low bandwidth of T1 circuits might make this use uneconomical. There is even discussion of ATM over 56K or 64K leased circuits, but, again, efficiency concerns argue against this. If one is running an IP or CLNP network, it is almost certainly more cost effective and possibly easier to run IP or CLNP directly over the leased telephone circuit than to overlay ATM where it isn't really necessary.

Where ATM might be headed

From this overview of the state of ATM and SONET, it is clear that it is too soon in the technology cycle for users to expect real multivendor interoperability. However, the needed specifications and standards are progressing rapidly. During the next two years, ATM and SONET networking will become much more interoperable than at present. Also, while it is currently premature to use ATM/SONET technology in mission-critical networks, such use will become feasible during the next two years as equipment vendors get through early development pains and network service providers widely deploy ATM service. Some view ATM as the only technology for the future and believe that it will be omnipresent. However, this appears to be unlikely to occur in practice. As noted above, the installed base of other LAN technologies continues to grow. Also, both Fiber Channel and the FDDI Follow-On LAN (FFOL) continue to be developed within ANSI. Moreover, ATM lacks certain desirable features of IP and ISO CLNP, for example dynamic routing. Hence it seems appropriate to ensure that internetworking protocols such as IP and ISO CLNP will operate well over ATM-based subnets. While network users should be tracking the development of ATM as an important future networking technology, they also need to thoroughly and objectively examine proposals and experiments that extend ATM outside of its original public switched network design objectives before deploying ATM in that manner.

References

[1] ATM Forum, "ATM User-Network Interface Specification, Version 2.0," ATM Forum, June 1992.

[2] Heinanen, J., "Multiprotocol Encapsulation over ATM Adaptation Layer 5," RFC 1483, July 1993.

[3] Atkinson, R., "Default IP MTU for use over ATM AAL5," work in progress, July 1993.

[4] Laubach, M., "Classical IP and ARP over ATM," work in progress, July 1993.

[5] Stallings, W., "Components of OSI: Broadband ISDN," *ConneXions,* Volume 6, No. 4, April 1992.

[6] Stallings, W., "Components of OSI: Synchronous Optical Network (SONET)," *ConneXions,* Volume 6, No. 4, April 1992.

[7] Clapp, G., and Zeug, M., "Components of OSI: Asynchronous Transfer Mode (ATM) and ATM Adaptation Layers," *ConneXions,* Volume 6, No. 4, April 1992.

[8] Blackshaw, R., "Components of OSI: Integrated Services Digital Networks (ISDN)," *ConneXions,* Volume 3, No. 4, April 1989.

[9] Leifer, D., "ISDN: Why use it?," *ConneXions,* Volume 4, No. 10, October 1990.

[10] CCITT Temporary Document 39, Melbourne, Australia, December 1991, WPXVIII/8 Meeting Report—Annex 3 (Part 2), "Text of WP XVIII/8 B-ISDN Recommendations," Appendix 5, Draft Recommendation I.150, "B-ISDN ATM Functional Characteristics."

[11] CCITT Temporary Document 39, Melbourne, Australia, December 1991, WPXVIII/8 Meeting Report—Annex 3 (Part 2), "Text of WP XVIII/8 B-ISDN Recommendations," Appendices 9 & 10, Draft Recommendation I.363, "B-ISDN ATM Adaptation Layer (AAL) Specification."

[12] ANSI T1S1.5 / 91-449, "AAL5—A New High Speed Data Transfer AAL," November 1991.

[13] IEEE Standard 802.6-1990, "Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)."

[14] CCITT Blue Book: "I.113 Vocabulary of Terms for Broadband Aspects of ISDN," 1988.

[15] CCITT Blue Book: "I.121 Broadband Aspects of ISDN," 1988.

[16] Laubach, M., "ATM for your internet—But When?" *ConneXions,* Volume 7, No. 9, September 1993.

RANDALL ATKINSON holds undergraduate and graduate degrees in Electrical Engineering and Computer Science from the University of Virginia. In a previous life he worked on software for programmable real-time controls. More recently, he has been working for the Naval Research Laboratory in high speed networking and high assurance computing. He is a member of the IEEE and the ACM and is active in both the Internet Engineering Task Force and the ATM Forum. He can be reached via e-mail at: `atkinson@itd.nrl.navy.mil`

DHCP—TCP/IP Network Configuration Made Easy

by J. Allard, Microsoft Corporation

Introduction

Each day new corporations, universities, and other organizations join the Internet to tap into its powerful resources. The result is explosive growth of the Internet, and more and more users with access to the broad base of information it offers. For many organizations, joining the Internet, or even creating private IP-based internets incurs significant administrative overhead. Putting aside the establishment and maintenance of the network connection(s), one of the biggest (almost daily) administrative tasks is *network configuration.*

The *Internet Engineering Task Force* (IETF) has designed the *Dynamic Host Configuration Protocol* (DHCP) to alleviate the network administrator's configuration burden when deploying TCP/IP in internets. DHCP centralizes TCP/IP configuration, manages the allocation of IP addresses, and automates much of the configuration process. DHCP was designed as an extension of the useful *Bootstrap Protocol* (BOOTP), already used to configure systems across internetworks.

One of the most significant enhancements offered by the DHCP protocol is the ability to dynamically configure workstations with IP addresses and associate a lease with the assigned addresses. DHCP leases offer an automated mechanism for the safe distribution and reuse of IP addresses in internetworks with very little administrative intervention. Further, the leasing policy can be tuned for a given network, providing compensation for the dynamics of network use. This article offers a look into this exciting new technology, explaining the basics of the DHCP protocol and how your network might best leverage it. See the documents identified in the References section for complete details on the DHCP protocol and configuration options.

Motivation

At minimum, each workstation in a TCP/IP internetwork requires configuration information before it is capable of accessing resources on the network. Typically, the minimum set of values necessary includes IP address, subnet mask, default gateway, and DNS address(es), although local policy may necessitate additional configuration information. Traditionally, this information is provided using arcane tools such as *ifconfig*, or by hand-editing configuration files.

Experience has proven that the average end-user on a network cannot be trusted to configure their own system(s) with correct network configuration information. In many cases, end users simply do not have the necessary access or skills required to successfully configure their system for network activity. Misconfiguration in TCP/IP networks is highly undesirable, as this often leads to network problems which can be difficult to detect without sophisticated network monitoring tools. Furthermore, the administrative overhead can decelerate and, in some cases, outright prevent the deployment of TCP/IP technologies in large enterprise networks. DHCP was designed to eliminate much of this overhead, allowing TCP/IP to be deployed easily in large networks with little end-user intervention or training required.

Beyond shifting the configuration responsibilities from end-user to network administrator, DHCP strives to make workstation configuration easier. Rather than building configuration profiles for all workstations on a given network, most administrators can develop profiles shared across multiple workstations sharing common characteristics. Finally, DHCP continues in the tradition of BOOTP to offer configuration support for diskless clients in need of network configuration information, as well as systems which require permanent addresses and configuration information.

DHCP concepts

DHCP is based on a client–server model. Using a simple UDP/IP based protocol, DHCP compatible clients communicate with DHCP servers and receive necessary network configuration information during system initialization. During this initialization process, clients are assigned IP address(es) with associated *lease(s).* The lease identifies the duration for which the client can safely use its dynamically assigned IP address. Under normal circumstances, DHCP clients should require no hand-configuration of network parameters whatsoever. Out of the box, DHCP-enabled systems should be able to join and communicate on TCP/IP networks without any administrative intervention.

In order to achieve this, the network administrator configures one or more DHCP servers with valid configuration information for requesting clients. In most cases, the administrator will configure a server with:

- A set of configuration parameters valid for all clients on the local IP subnet
- A pool of valid IP addresses to assign for these clients
- The duration of the leases to be offered by the server

The parameters assigned during the DHCP initialization are referred to as DHCP *options.* Several common options (mainly extracted from the host requirements documents) are documented in the *DHCP Options and BOOTP Vendor Extensions* document [1]. The options document also describes the mechanism used to extend these options to carry vendor specific options without fear of conflict.

Relay agents may be used to allow DHCP servers located on one IP subnet to service configuration requests from remote subnets. The relay agent simply forwards requests from local DHCP and BOOTP clients to remote BOOTP and DHCP servers. DHCP server responses are similarly relayed through these agents to the clients. Relay agent software is available for many IP routers as well as stand alone daemons which can be run on workstations. The document *Clarifications and Extensions of the Bootstrap Protocol* [2] discusses the behavior of relay agents.

The protocol

During initialization, DHCP clients broadcast a DHCPDISCOVER message to their local subnet over the well-known BOOTP server port. (UDP Port 67 as defined by the Assigned Numbers RFC.) BOOTP relay agents may optionally extend the radius of DHCPDISCOVER messages by forwarding them on to DHCP servers located on remote subnets as well. All participating DHCP servers within the broadcast area which have a valid configuration for the requesting client respond on the BOOTP client port (UDP Port 68) with a DHCPOFFER message (via unicast whenever possible). Local servers respond with a local broadcast or unicast, remote servers respond via the relay agent which forwarded the initial DHCPDISCOVER message. The DHCPOFFER message(s) contains an IP address and configuration valid for the requesting client. As the DHCP client collects configuration offerings, it enters the *selecting* state (see Figure 1).

At this stage of the protocol, one or more DHCP servers may have outstanding offers for the client. When an offer is made, the server should temporarily reserve the offered address from its pool preventing it from accidentally offering it to another requesting client. The client, enters the *requesting* state by choosing one of the configurations offered and indicating its choice using a DHCPREQUEST message.

continued on next page

DHCP *(continued)*

The DHCPREQUEST message contains (at minimum) the *server identifier* option which identifies the DHCP server whose configuration it has selected. The request is delivered to the same DHCP servers which received the client's initial discover message, again via local broadcast and/or relay agents. Since the client's DHCPREQUEST message contains the identity of the DHCP server whose configuration it has chosen, any other servers which have offers outstanding for the client can safely return the offered address to the free address pool.

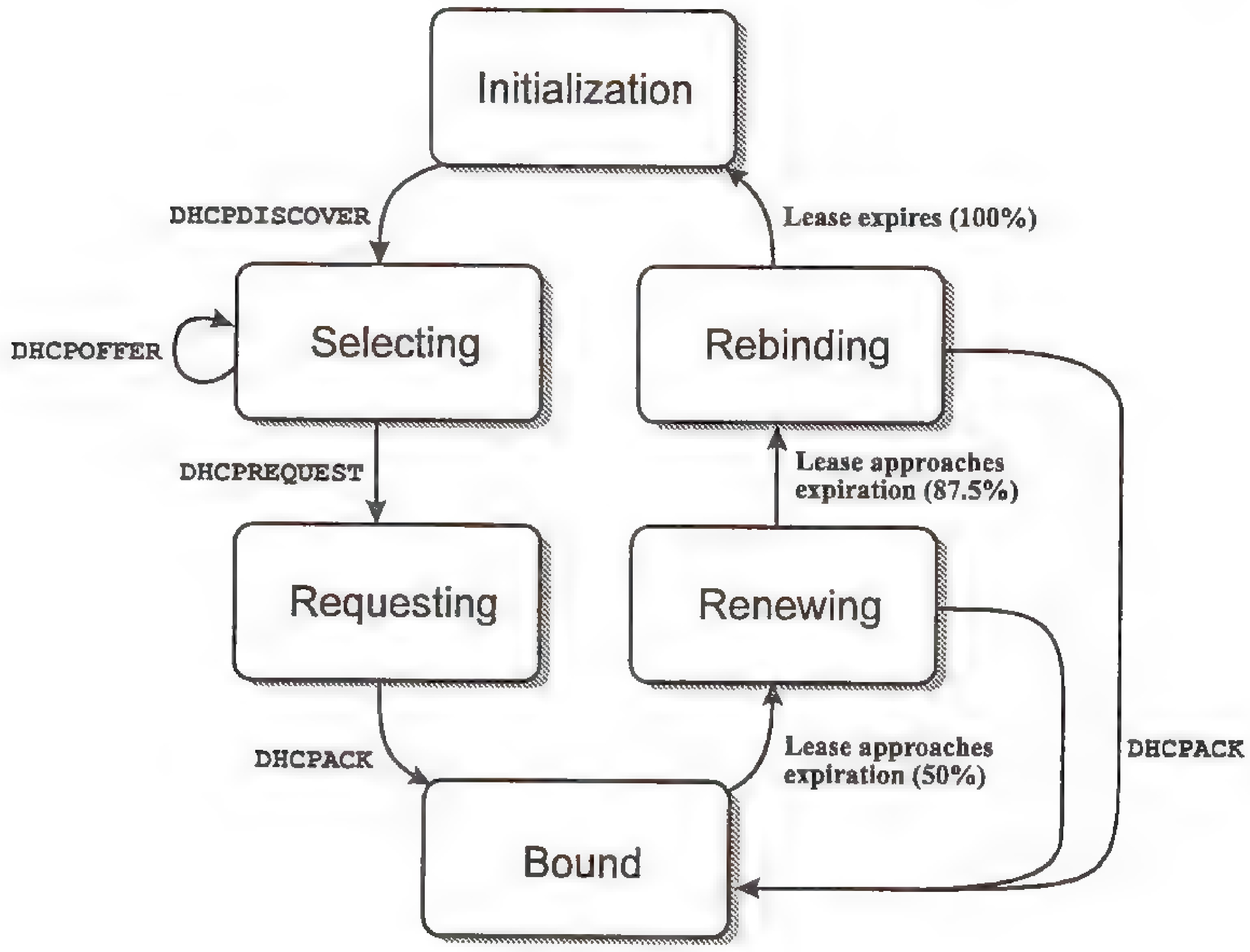

Figure 1: DHCP Client State Transition Diagram

Barring catastrophe (natural, network, electric or otherwise), the selected DHCP server builds a DHCPACK (DHCP acknowledgment) message which contains the address negotiated during the discovery process (now protected from re-assignment), a valid lease for the address (based on local policy), and the network configuration parameters for the client. Upon receipt of this acknowledgment the client enters the *bound* state, now able to participate on the network and to continue its normal system initialization processing. As an optimization, clients which have local storage available will generally commit (at least) the received address for use during subsequent initializations.

When a client has been active for one half the duration of its lease, the client enters the *renewing* state, and attempts to renew its lease with the server which it received the configuration from. This is achieved by sending directed DHCPREQUEST message to the server indicating the address it has been assigned. If lease extension conforms with local policy, the server responds with a DHCPACK containing the new lease as well as valid configuration parameters for the client. If the client is able, it should update its current values for any parameters to those received in the DHCPACK as they may have been updated by the local administrator. Once a client's lease is updated it returns to the *bound* state. In the *renewing* state, the client must be prepared to release its address immediately in the unlikely event that the server responds with a DHCPNAK (DHCP negative acknowledgment) message. The DHCPNAK message will likely be used only in rare situations. This message was added to the DHCP protocol to allow for a mechanism to inform clients that they have received incorrect configuration information forcing them to release their address and re-acquire new information.

Should a directed renewal attempt fail, the client will continue directed attempts until 87.5% of the lease time has expired. If no directed attempt has succeeded by this point in time, the client enters a *rebinding* state, attempting to acquire a lease extension from any DHCP server. This is achieved by broadcasting DHCPREQUEST messages such that all servers within the client's radius have equal opportunity to renew the client's lease. Any server that wishes to extend the lease does so by responding with a DHCPACK message containing the extended lease and updated configuration parameters. Once again, DHCP servers may respond with a DHCPNAK message in this state, forcing the client to release the current configuration and return to the *initialization* state.

Should the lease expire completely, or if the client receives a DHCPNAK message at any stage of the renewal/rebinding process, the client is responsible for immediately discontinuing the use of the TCP/IP network on the address which has expired. If the client TCP/IP software is able, it may re-enter the *initialization* state and attempt to acquire a valid address with a new lease. In the case where a DHCP client has multiple network interfaces installed, the above protocol is followed over each interface for which DHCP configuration is desired.

Anatomy of a DHCP message

The DHCP message format is nearly identical to the BOOTP message format. This design was both convenient and practical. Using the same message format allows BOOTP server implementations to be rapidly enhanced to service both BOOTP and DHCP clients. Moreover, BOOTP relay agents can be used to forward DHCP requests across subnet boundaries. With the exception of the options section, all of the fields in the DHCP message are of fixed length:

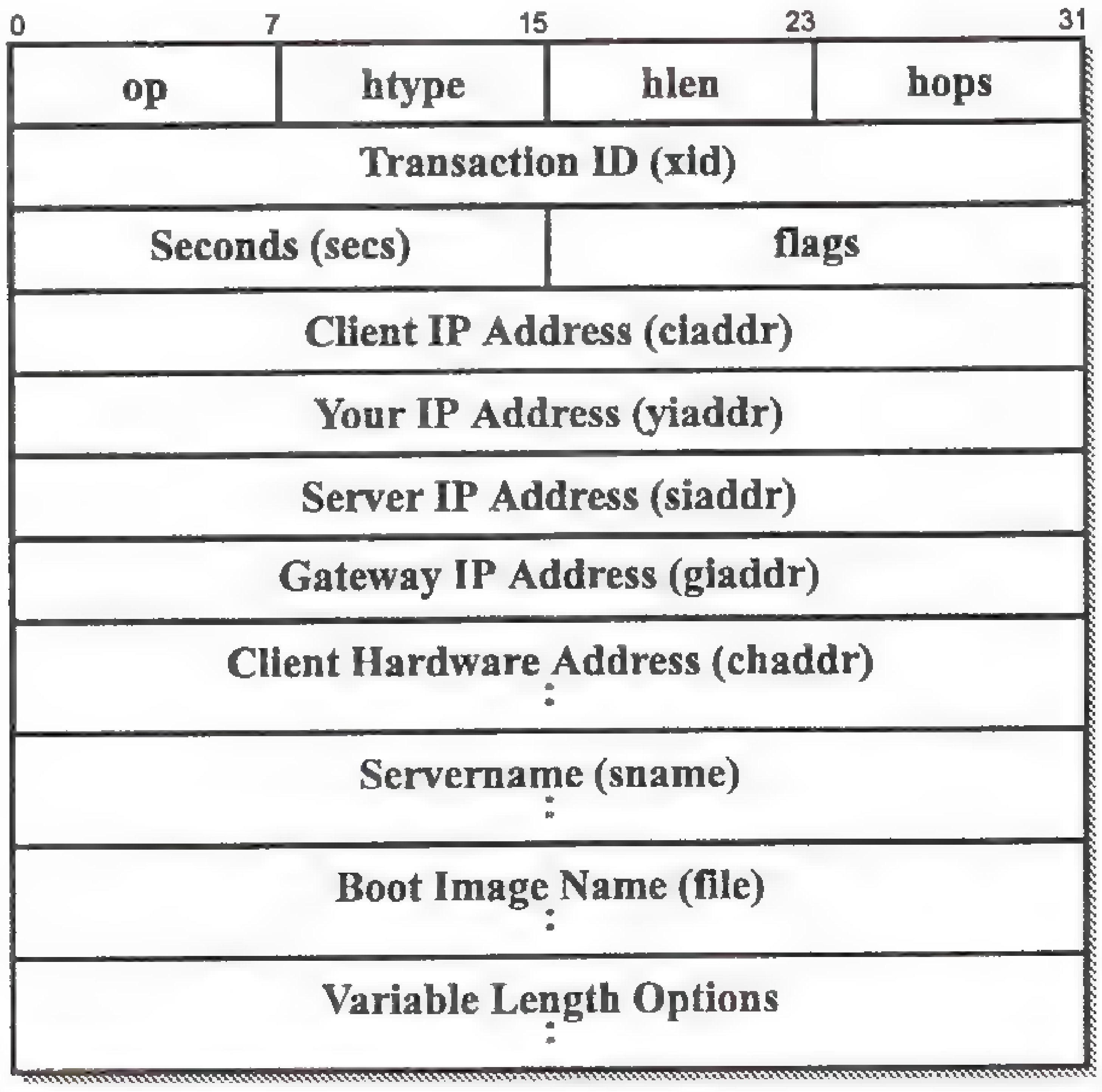

Figure 2: The DHCP Message Format

The *op* field is used to specify whether the message is a boot request or boot reply. In all DHCP interaction, the *op* field is set to BOOTREQUEST (1) by DHCP clients and BOOTREPLY (2) by DHCP servers. The formal DHCP message type is specified in the DHCP Message Type Option (described in the next section). The *htype, hlen,* and *chaddr* fields are used by the client to specify a client's hardware address prior to address allocation. *chaddr* may be used by DHCP servers or relay agents which are capable responding to a client's hardware address.

continued on next page

DHCP *(continued)*

The *flags* field accounting for two previously unused octets in the BOOTP protocol now defines the high-order bit of this field to specify whether or not clients are capable of receiving unicast responses prior to address allocation. The remaining low-order bits of this field are reserved, and must be set to zero. The *hops* and *secs* fields may optionally be used by relay agents during the initialization process. The *xid* field is a random number generated by the client during discovery which provides an association between client requests and server responses by remaining in all subsequent DHCP messages. The *sname* and *file* fields may be used by BOOTP or diskless clients, but if unused, may be overloaded to carry DHCP options by the use of the Option Overload DHCP option.

The *ciaddr* field is used to request a specific IP address by clients which have been previously allocated an address that they wish to keep. The *yiaddr* field is filled in by DHCP servers, containing the IP address offered to a specific client in DHCPOFFER and DHCPACK messages. Finally, the *giaddr* field is used by relay agents and servers to properly deliver DHCP messages across subnet boundaries. DHCP clients always set this field to all zeros. When a relay agent forwards on the request to a DHCP server, it fills in the *giaddr* field with its own IP address, allowing the server to direct responses back to the relay agent, which in turn can forward responses to the requesting client.

DHCP options

DHCP *options* are included in the DHCP message following the fixed length fields detailed above. To remain compatible with BOOTP implementations, the first four octets of the options field contains the "magic cookie" value defined by RFC 1395 (99.130.83.9) followed by individual DHCP options. Individual DHCP options are encoded in the following format:

Option Code	**Length**	**Data...**

Figure 3: DHCP Option Encoding

An option code is a single octet value which provides a context for the data which follows. Options codes between 0 and 127 inclusive and 255 are "standard" options, specified in [1]. Following the code is an octet of length information, indicating the number of octets of data which follows for the given option. The actual data or the value of the option follows the length octet.

Options can be packed together, or may be aligned on word boundaries using trailing Pad Options (option 0) which are simply ignored by the client. The DHCP server may include as many options as may fit into a DHCP message, and follows the final option with the zero-length End Option (option 255) to indicate the end of option processing for the client. Finally, the server is required to fill the rest of the DHCP message with zeros.

Most options specify network transport configuration values such as subnet mask, DNS server address(es), MTU and so on. Others are used by the DHCP protocol and are required in many messages. For example, the DHCP message type is DHCP option number 52, has a length of one octet, and has several values defined to specify the type of message being sent. For example, a DHCPREQUEST message would contain DHCP option number 52 with a value of 3 (DHCPREQUEST message type):

Code	Length	Data
52	1	3

Figure 4: A DHCPREQUEST Option

Currently, the DHCP Message Type option defines 7 values encoded as follows:

1	DHCPDISCOVER	5	DHCPACK
2	DHCPOFFER	6	DHCPNAK
3	DHCPREQUEST	7	DHCPRELEASE
4	DHCPDECLINE		

Figure 5: DHCP Message Type Option Values

Some options such as the DHCP Message Type option are fixed length, their length value will always be the same as specified in [1]. Other options such as hostname are variable length, and still others are required to be a multiple of some value. Regardless of the message type, its length is *always* specified in the required length field. The encoding of the data length allows both variable length and fixed length options to be decoded by DHCP clients, even if they are incapable of processing or identifying a particular option type.

DHCP addressing policy

Three allocation policies are defined for DHCP address distribution: manual, automatic, and dynamic. All three policies use the same fundamental client–server discovery and delivery protocol, but are managed differently by the network administrator at the DHCP server.

Manual allocation refers to the policy supported by most conventional BOOTP implementations. This scheme allows network administrators to centralize the configuration information for individual workstations on the network at the DHCP server. DHCP is simply used as a mechanism to deliver the network address and other configuration options to the workstation. The response generated by the DHCP server(s) for a given workstation is predetermined by the administrator and is based on the *unique client identifier* option specified by the workstation during DHCP initialization. Generally, this unique client ID will be the MAC-layer address (e.g., Ethernet address) of the interface to be configured. There is a one-to-one mapping between the unique client identifier offered by the client and the IP address returned to the client by the DHCP server. Other configuration options may be specified for the client, or the client may simply inherit the configuration options for other clients on the same subnet. In any event, it is necessary for the administrator to provide the unique client ID/IP address mapping used by the DHCP server. Any IP addresses assigned in this fashion are unavailable for either automatic or dynamic allocation by any DHCP servers for clients with any other unique client identifier.

Automatic allocation is similar to manual allocation in that there exists a permanent mapping between a workstation's unique client identifier and its IP address. However, in automatic allocation, this mapping is created during the initial allocation of an IP address. Under automatic allocation, the DHCP server software assigns an IP address to a DHCP client during its first DHCP initialization, and then retains the mapping for future requests to ensure that future requests will result in the same address being distributed to the client.

continued on next page

DHCP *(continued)*

The IP addresses assigned during automatic allocation come from the same pool as dynamic addresses, but once assigned cannot be returned to the free address pool without administrative intervention. Both automatic and manually assigned addresses are considered to have permanent, or infinite, leases.

Dynamic IP address allocation takes advantage of the most interesting aspects of the DHCP protocol: dynamic network address allocation and reuse. Dynamic assignment allows a workstation to be assigned an IP address from a pool of available addresses and be given a non-infinite lease for which the address is valid. As the lease nears expiration, it is the responsibility of the client to renew its lease if it is still in use. Pursuant to local policy, a dynamic address can be held indefinitely provided that the client continues to renew its lease before the lease formally expires.

It is more likely that dynamically allocated addresses will be freed and returned to the pool of available addresses. This may happen when a client is decommissioned, or moves from one subnet to another. The address may also be returned when the user of the workstation powers down their system for an extended vacation. In any case, once an address is returned to the free pool, it is "up for grabs," that is, the IP address may be reused by the DHCP server to configure a client which does not currently hold a valid IP address. Addresses are returned to the pool when the client specifically releases the address by sending a DHCPRELEASE message, or when the address lease expires.

As much as possible, the dynamic allocation policy should attempt to assign the same network addresses to clients during re-initialization. This implies that a strong DHCP server implementation should discourage reuse of addresses whose leases have expired until the unused and released addresses are exhausted. This will typically be accomplished by applying a least-recently-used algorithm to the "recycled" addressed in the pool when making a new assignment, and by retaining the IP address/unique client ID mapping at the server.

DHCP server administration

In most TCP/IP environments, all of the workstations on a particular IP subnet share common configuration information. For example, the workstations on a given subnet generally share the same subnet mask, default gateway, and DNS server configuration information. In many cases, the only configuration values specific to any particular workstation on a subnet is its IP address. To this end, the network administrator will begin DHCP server configuration by creating a common configuration profile that is shared among all workstations for a given subnet. Associated with the configuration profile is a range of valid IP addresses which may be assigned, and the policy by which this range will be administered. In most situations, dynamic allocation will be the most desirable policy to enforce.

Determining the configuration parameters for a given subnet is fairly straightforward. The network administrator has likely configured systems on this subnet before and is familiar with the local topology and policies. The more difficult aspects of configuration include dealing with non-DHCP clients on the subnet and determining the appropriate lease for reuse of network addresses. To take care of non-DHCP clients, a server implementation should be configurable to exclude any statically assigned (i.e., hand-configured) addresses from the pool that it allows the DHCP server to manage.

Profiles for diskless workstation/X terminal BOOTP clients will also need to be built, or can continue to be managed by existing BOOTP server(s) (in which case the DHCP server would exclude these addresses from its pool as well).

In order to determine fair address leasing values, the network administrator will have to consider the frequency of network interface failures, decommissioned systems, and subnet "jumps" (office moves, laptop users, etc.). All of these activities will cause assigned IP address leases to expire at the DHCP server(s) (if they are not explicitly released), causing the IP addresses to be returned to the address pool for reuse. In an environment where systems have a high frequency of subnet jumping, it would be preferable to assign a short lease time (for example, two weeks) so that addresses assigned to systems which leave the subnet can be reused quickly by new systems joining.

Perhaps the most important factor in the lease duration determination process is the ratio between connected systems and available IP addresses. If 40 systems are sharing a class C address (254 available addresses), then the demand on an address for reuse is low. Configuring a rather long lease time (e.g., 2 months) in this situation would be safe. On the other hand, if 230 system share the same pool, then reuse availability becomes a much greater issue and a lease time measured in days or maybe weeks would be more appropriate. In most environments, this will probably require some tuning to get it right.

Another administrative consideration is that of redundancy. Imagine a network in which a single DHCP server responsible for 230 systems on a particular subnet goes down. Is it acceptable for this failure to prohibit any of the systems from initializing? For most network administrators, the answer would be a solid "no." It is desirable to make use of redundant servers to safeguard against situations like this. Unfortunately, the DHCP protocol does not define a mechanism by which DHCP servers can cooperate to ensure the singularity of assigned addresses. It is therefore necessary for the network administrator to manually partition the available address pool amongst servers in order to protect from duplicate address assignments.

A common scenario exists where a single, local DHCP server maintains configuration information for two subnets. As as example, consider two connected subnets, each with its own DHCP server. On each of the DHCP servers the network administrator configures the system to maintain 70% of the address pool for local clients and the remaining 30% of the pool for clients from the remote subnet. The administrator then configures a relay agent to deliver requests between the two subnets. This results in the local DHCP server being used by local clients most of the time (many DHCP client implementations will likely choose the first offer they receive). The remote DHCP server will only assign addresses to clients on the other subnet when their local server is unavailable, heavily loaded, or out of addresses. With the low cost of manual pool partitioning, this method can be extended beyond the 2 subnet scenario to ensure even greater availability. New DHCP extension protocols are under development to assist in the automatic partitioning of address pools across multiple servers.

Future directions

DHCP marks a starting point for truly dynamic internetworking with TCP/IP. Future efforts will be required to consider the DHCP protocol in design and implementation, and will benefit by its existence. In the short term, two immediate problems will need to be addressed by the Dynamic Host Configuration Working Group: a DHCP server–server protocol, and dynamic naming with the *Domain Name System* (DNS).

continued on next page

DHCP *(continued)*

The DHCP client–server protocol takes great steps to make network configuration easier for administrators, but it does not provide a simple means to solve redundancy and effective address reuse by multiple servers. In a production environment, most administrators will want to have more than one DHCP server available for servicing client requests for a given subnet. Without a server–server protocol, the administrator is forced to divide the address pool into disjoint chunks, distributing independent ranges of addresses to all "cooperating" servers. This is necessary to provide reliability in the event that any one server becomes unavailable. However, since the DHCP servers are not cooperating with one another, a client can only use a single server to renew a lease (specifically, the server it acquired the address from). Moreover, the address pool may not be divided fairly, and does not protect from address exhaustion on a given server. Development of a DHCP server–server protocol is under cosideration which will allow for more reliable and fair service in multi-server environments.

The dynamic naming/DNS problem is an interesting one. Although DNS is extremely useful in allowing users to provide "friendly" names for resources on a network, its administration and configuration is extremely static. With the advent of DHCP, it is possible that a host which has an IP address *X* today, will be assigned an IP address of *Y* tomorrow. Since there is presently no standard protocol to dynamically update DNS servers when IP address information is updated, DNS naming conflicts may crop up when administrating IP addresses dynamically via DHCP.

The DNS problem primarily affects systems which extend services to users of the network. For example, a server which acts as an anonymous FTP server or as an e-mail gateway will likely require that users be able to contact it by name using DNS. In these situations, DHCP will likely be used in either "automatic" or "manual" mode for some time, assigning permanent lease addresses to clients so that DNS naming conflicts will not occur. On the other hand, workstations which are not required to be registered in the DNS namespace are able to take advantage of DHCP dynamic assignment without fear. Many environments will fall into this category and will tailor their DHCP address allocation policies to suit their individual networks.

J. ALLARD is the program manager for TCP/IP technologies at Microsoft Corp. His group is focused on providing TCP/IP solutions for the Microsoft Windows and Windows NT operating systems. A former system administrator, much of his focus in product development is making advanced networking easier to install, configure and use. He has been involved in several network standards efforts in the IETF such as DHCP, and was an active contributor to the Windows Sockets API, a standard for network programming under Microsoft Windows. He received his B.S. in Computer Science from Boston University. He can be reached on the Internet as:
`jallard@microsoft.com`

Conclusion

Whether administrating a local IP network, or connecting to the global Internet, DHCP provides a simple and reliable means for network administrators to centrally configure networked workstations. The IP address distribution policy allows addresses to be assigned automatically and reused without worry of misconfiguration or duplicate addresses on the network. With technology in place to connect "out-of-the-box" systems to complex internetworks and the global Internet, administration is no longer a barrier to the information and resources that TCP/IP applications and internetworking provides.

References

[1] S. Alexander and R. Droms, "DHCP Options and BOOTP Vendor Extensions," Internet Draft, June 1993.

[2] W. Wimer, "Clarifications and Extensions for the Bootstrap Protocol," Internet Draft, December 1992.

[3] J. Reynolds, "BOOTP Vendor Information Extensions," RFC 1395, January 1993.

[4] R. Droms, "Dynamic Host Configuration Protocol," Internet Draft, June 1993.

Expanding International E-mail Connectivity: Another Look

by John Klensin and Randy Bush,
Network Startup Resource Center

Collaboration

Unlike the situation of thirty years ago, when almost all important scientific work occurred in Western countries, the scientific community is becoming increasingly international. Important work and areas of study occur all over the world. Collaborations and ability to access sources of data and other resources are increasingly important to scientific progress. In many fields, we see more and more inter-institutional collaborations on research and papers that draw on the strengths of each of these institutions. Exchanges of ideas and collaboration and review of proposals should not be limited to one country, or even to developed areas. Especially in such areas as the health and social sciences and in all of the various fields that study "global and environmental future" issues, participation of scientists in developing areas has become crucial. This is true whether the scientists themselves are indigenous to, or visiting in, those areas; indeed, as the community becomes more international, the distinction between the two is gradually becoming less clear.

Local policies often reinforce the trend toward collaborations that require strong communications links. For example, as it becomes harder to move biological samples or historical artifacts across international boundaries, it makes increasing sense to do analyses and evaluations within the country of origin, then make the data available to both both domestic and remote parties. To the degree that relationships involving local and remote scientists and institutions become permanent and stable, the benefits tend to spread with improved research and more open and tolerant relationships with regulators and those not initially involved.

E-mail: A critical tool

Communication facilities based on computer networks, especially the low-level ones such as electronic mail, have become critical to these types of collaboration. The post is simply too slow to permit real time interaction, and fax, while faster, does not lend itself well to group communication, much less true collaboration on shared materials. The more the network connection infrastructure can be opened up, the more scientists can participate in international efforts on the basis of interests, skills, and knowledge, rather than based on where they happen to live or work.

Similarly, data gathered or prepared in remote locations often must be transferred elsewhere for evaluation or analysis. The examples usually cited are climate or rainfall data, but similar issues arise with health and nutrition status, mortality and disease figures, and even certain types of economic statistics. Without computer networks, the options are to mail machine-readable media (often unacceptably slow and unreliable) to send the data via fax (typically requiring rekeying or the use of OCR techniques, which are not completely reliable), or long-distance international data calls at great expense.

Many Latin American, Caribbean, and African universities tell of invited US scientists who would not come for sabbaticals or extended research visits because they would not have e-mail access to their colleagues. The problem of being without network facilities is not merely one of keeping up correspondence, but of being isolated to the point of ineffectiveness if reliable e-mail is not available.

continued on next page

International E-mail Connectivity *(continued)*

Infrastructure fosters collaboration

On the other hand, when networks are available, previously-unanticipated collaboration seems to come into being almost spontaneously. Again, the underlying causes seem to involve a latent demand that remains latent as long as joint work requires either the disruption of waiting for the post, the continual retyping of texts transmitted by post or fax, or the need to secure large budgets and approvals for extensive international travel.

When computer networks are available, people quickly become comfortable with using them, the collaboration seems to happen often and quickly in many disciplines and work groups, and needs only a little bit of outside stimulus to get it started in others.

These patterns in the scientific community have been paralleled by activities in various agencies and organizations concerned with development or assistance. Groups in one country need to communicate with those in another, and, when they can be made available, electronic mail and computer-assisted communications have significant advantages over other approaches.

Trends

The demand from both communities has been present for some years as made evident by the "how do I access a network to stay in touch with colleagues at home while I'm in the field or to work with colleagues in remote locations from my home institution," inquiries which appear at very frequent intervals on popular network mailing lists and news groups. In what we might think of as first-generation low-end wide area networking, the response was "call home": connections from a terminal in the field to a centralized computer. The calls might be made with modems over international telephone connections or remote-connection PPSDN links (usually X.25), but the essential communication pattern was remote login to a centralized computer that hosted what was, in reality, a centralized e-mail system. Gradually that type of arrangement became somewhat less decentralized: a single central computer that everyone dialed into was replaced by regional central computers with the same type of arrangements and some way of communicating among themselves.

As computer costs dropped and modem technologies improved to permit data communications over low-quality lines at higher speeds and plausible costs, opportunities arose for true computer-computer connections, with people receiving and composing mail on their local systems, rather than trying to type while connected to remote locations. Of course, those opportunities have been taken advantage of, but often in a way that may inhibit positive long-term network developments.

Drawbacks

Organizations with a need to communicate with subsidiaries or collaborators can now establish single-purpose polled or dialout arrangements, typically using FidoNet or UUCP technology, that link the components of that organization, or the collaborators in a particular project, with each other. In any given situation, this may be reasonable and the arrangements can be established with a minimum of fuss. Unfortunately, there are also negative effects:

- Participants in one activity tend to become isolated from non-participants and participants in other activities.
- While inter-country communication may be facilitated, intra-country communication may be frustrated: either made impossible altogether or forced through very remote gateways.

- While two separate private arrangements may be cost-effective, the third one rarely is and there are usually major advantages in not starting the second. If a single organization can afford one polled international mail exchange a day, better service for everyone can typically be arranged for the same costs by sharing resources and arranging multiple exchanges. If the user base is wide enough and can be expanded without regard to project boundaries, it has been shown time and time again that a minimal networking channel will quickly build a user base which soon fills that channel. Even in less-developed areas, the perception of value rapidly builds to the point that the user base then manages to find its own funding to continually increase the available bandwidth.
- Private arrangements tend to satisfy, and then hide, demand. Resources being invested that could contribute to higher-quality connections for an area do not appear when needs or market surveys are performed. This is especially critical in situations in which the precondition for building a network infrastructure is the ability to demonstrate that the demand and users exist. This demand often does not arise until networks are actually seen to be in place and working, and people involved in private network arrangements might otherwise make major contributions to it. Looked at differently, private arrangements tend to be examples of classic cream-skimming behavior: the needs which can be most easily financed are met, making it more difficult to meet needs—possibly more serious ones—that are less readily financed.
- Network arrangements set up for a particular project tend to collapse when that project ends, leaving people who had learned to depend on a certain level of communications and connectivity without it.

Experts versus users

The decision to install a private network—or, more often, simply a private star-type mail polling arrangement—often results from lack of understanding or consideration of long-term implications. Many of the people who are normally considered experts can make poor guesses and give poor advice if cost and technology tradeoffs are radically different than they are in areas with established telecommunications and network infrastructures. This situation prevails, almost by definition, in many less developed areas. Although the reasons may be different, it is also prevalent in many areas of Eastern Europe and other portions of the former Soviet area of influence. A very similar situation occurs in the US K–12 arena, where the cost of a single phone line can be a major administrative obstacle.

Even when networks develop within an area, without significant impetus from "outside," the user community that drives the installations may turn out to be the wrong one in the long term. For example, the history of starting networks and network connections has predominantly rested in computer science and computer technology-oriented departments, businesses, and other data communities. There is some history of these communities constructing networks for their own use and then using various mechanisms—costs, perceived complexity, or lack of user support to then hoard the resource.

Then, when computer *user* communities—scientists, educators, or the general public—for whom the computer is a tool for communication or computation (but not an interesting device in its own right) need access to networking technology, they often need to start over.

continued on next page

International E-mail Connectivity *(continued)*

At the same time, those user communities are much larger and, in many places, represent the largest potential user community. They may ultimately have access to greater political, cultural, and financial resources than computer technologists.

At the same time, the community which is more computer-oriented can, and typically has, managed to establish communications when they see it as sufficiently important. While they may not understand optimal approaches for a given area, they usually have access to sufficient information to make something work. Other scientists and organizations have often had less success, since they do not have the technology readily available and may believe (or "know") many things that are untrue, such as the need to focus computer networks around mainframe systems or centralized remotely-accessed hosts or the requirement for very high-bandwidth circuits to do anything useful. These misperceptions are encouraged by publicity releases from vendors of high-end systems and by a media focus on the "next generation."

These problems, and the behavior patterns that cause them, are not limited to international development or developing countries. For example, in the US K–12 (kindergarten through 12th grade educational) community, few of the innovative teachers, the potential initial users who perceive the benefits in advance, have the political and economic power to affect their networking destiny. Network inertia and data hoarding are rife in the administrative infrastructure, often leading to either no connections at all or to restricted private single-function networks. Theories of technological trickle-down are routinely cited but regularly disproven in the data networking arena.

A new approach

All in all, it may have become a little bit too easy to set up a "network." Maximum effectiveness in network-building requires a different approach in which we phase out remote dialup arrangements in favor of "hosts" co-located with the users and where possible phase out—and stop creating—private arrangements in favor of shared connections and infrastructure.

We need to improve our structure of information about who is interested and what is already operating in a particular area so that, if there is will to cooperate, every effort in a particular area reinforces every other effort and strengthens the links to the outside. This, in turn, leads efforts down the path toward regional backbones as the most effective mechanism for providing adequate bandwidth through, and out of, the countries and regions. Interested parties can best leverage their own needs into effective networks if they have information about other interests and activities. If existing sites are not inclined to cooperate with new ones, the best solution is to simply develop parallel infrastructure, gradually leaving them out. That, of course, requires the same databases, training, and information as would be the case if there were no existing connections to the area.

But, if we fail to move in ways that consolidate efforts and lead to better communications and interconnections within areas and between projects and disciplines, we shall see increasing intercommunication and connection difficulties among people and groups who "have e-mail" or are "connected to networks."

References

[1] Ezigbalike, I. Chukwudozie and Ochuodho, Shem J., "E-Mail for Developing Countries—What They Never Tell You About It," shem@minster.york.ac.uk.

[2] Landweber, L. H., "International Connectivity, Version 7," Spring 1993, lhl@cs.wisc.edu.

[3] Lottor, Mark, "Internet Growth (1981–1991)," RFC 1296, January, 1992.

[4] Solensky, Frank, "The Growing Internet," *ConneXions,* Volume 6, No. 5, May 1992.

[5] Marine, A., "How Did We Get 727,000 hosts?" *ConneXions,* Volume 6, No. 5, May 1992.

[6] Press, L., "Relcom: An Appropriate Technology Network," Proceedings of INET '92: The International Networking Conference, Kobe, Japan, June, 1992.

[7] Goldstein, S. & Michau, C., "Convergence of European and North American Research and Academic Networking," *ConneXions,* Volume 5, No. 4, April 1991.

[8] Mike Lawrie, "Research and Academic Networking in South Africa," *ConneXions,* Volume 5, No. 8, August 1991.

[9] Steve Neighorn, Randy Bush, and Jeff Beadles, "Profile: RAI-Net," *ConneXions,* Volume 6, No. 5, May 1992.

[10] Mark Bennett, "Electronic Mail in Zambia," *ConneXions,* Volume 6, No. 9, September 1992.

[11] Press, L.,"INET '92: The Start of Something Big," *ConneXions,* Volume 6, No. 12, December 1992.

[12] Dyson, Esther, "Eastern Europe Trip Report, Release 1.0," May 31, 1990, EDventure Holdings, New York, pp 1–30.

[13] Kessler, Jack, "Europe—at least—discovers the users," *ConneXions,* Volume 7, No. 7, July 1993.

JOHN KLENSIN holds an S.B. and Ph.D. from MIT. He is director of the INFOODS Secretariat for the United Nations University and was until recently Principal Research Scientist at MIT. He has worked on or led major projects in statistical and scientific database management, interchange of very complex data, information location and retrieval with uncertain classification, data analysis and modelling, and the impact and influence of communications. He has tried to use computer networks to enable applications and non-expert users since the early days of the ARPANET, and has occasionally succeeded. He is a member of the Internet Engineering Task Force and chaired the recent working group on extensions to the SMTP protocol. He is also a member of ACM and immediate past chair of its Standards Committee, IEEE, the American Statistical Association, and the International Association for Statistical Computing. He can be reached via e-mail as: Klensin@INFOODS.UNU.EDU

RANDY BUSH is a compiler netware, and tools hacker, and too often a software engineering manager. Residing in Portland Oregon US, he is currently a software architect at Olsen and Assoc., Zurich. He has been a user and occasional implementor of networking for a few decades, and is a member of the Modula-2 language committees and other lost causes. He has been involved in in integration of appropriate networking technology in the developing world for over four years, using FidoNet, UUCP, and TCP/IP. His e-mail address is: randy@psg.com

Dr. Klensin and Mr. Bush are the principals in the Network Startup Resource Center, an effort to provide technical assistance to people developing computer networks, especially low-end ones in developing areas. That effort, and the development of this article, are supported in part by NSF Grant NCR-9216064.

The NREN Moves Towards a National Information Infrastructure

by Mike Roberts, EDUCOM

Introduction

The new White House staff, caricatured in the Washington press as "twenty-somethings," have brought an activist technology agenda with them. Following a media blitz in February featuring Bill and Al in the Silicon Graphics cafeteria proclaiming a future of "Information Superhighways," more than a dozen bills have been introduced in Congress dealing with information infrastructure, networks, and telecommunications.

The combination of growing political interest with explosive growth of use of the Internet has brought new players to the policy arena, most notably the regional Bell operating companies. Their March policy statement joined others in endorsing the Clinton vision of a national information infrastructure but added a sharp edged message that the role of government was to conduct basic research and leave the rest of the work to the private sector.

HR1757

Representative Rick Boucher of Virginia, Chairman of the House Science subcommittee, completed an active spring of lawmaking when his bill was reported to the full House of Representatives at the end of June. HR1757, the *National Information Infrastructure Act of 1993,* contains two main themes. The first is to push federal research and development support in the direction of networked applications. This reflects a concern in Congress that too little of the $70 billion annual federal R&D budget generates results that are useful to the private sector in creating jobs or increasing industry competitiveness. The bill authorizes more than $800 million for applications, including specifically the areas of education, health care and libraries.

A Key Role for The Internet

A second theme of the legislation is to update and clarify federal policy towards the Internet. Prior work, spearheaded by Al Gore during his Senate years, defined a *National Research and Education Network* that was to achieve gigabit bandwidth by 1996. Ever since passage of the *High Performance Computing Act of 1991,* written by Gore and his staff, the NREN has been the object of great confusion as to what it is and whom it is to serve. The 1993 bill repositions the NREN as a federal program rather than a network, and specifically identifies the Internet as a means for providing wide network connectivity among all citizens and as a base for federally sponsored applications. Section 305 of the bill authorizes NSF to foster networking "in local communities which will connect institutions at all levels, libraries, museums, and State and local governments..."

Testbeds

HR1757 also attempts to deal with the tricky issue of when and how federal networking initiatives ought to transition to the private sector. It creates a new program category of "testbed networks" whose purpose is to develop and demonstrate advanced networking technologies and to provide connections to such advanced services when they are not available from commercial sources. It goes on, however, to require that testbed networks are not to be used for purposes other than the federal program of which they are a part, nor to provide services that are commercially available. The intent of the language is to deal with the gray area between technology that looks promising in the lab and its adoption in commercial telecommunications service.

The bill is expected to pass the House during July and be taken up by the Senate in September. If the pattern of the previous legislation is repeated, it should get to the White House for Presidential signature in early October.

MIKE ROBERTS is Vice President for Networking at EDUCOM, a 600 member association of colleges and universities with common interests in information technology. The EDUCOM Networking and Telecommunications Task Force, a group of sixty universities and corporations, of which he is the staff director, has been active in planning and advocacy for the NREN. E-mail: roberts@educom.edu

InterNIC Directory and Database Services

Introduction

Members of the NSFNET community have recently expressed concern about the continued availability of the WHOIS directory records that have previously been available through WHOIS but that do not contain point of contact information. The NSF InterNIC team recognizes the importance to the NSFNET community of this information and, therefore, will ensure that this information is available to the community. The continued maintenance of this information in a centralized fashion is inconsistent, however, with the longer term need for a distributed directory architecture in which data is co-located (logically, if not physically) with those who have the knowledge to keep the data accurate and up-to-date.

WAIS and X.500

Applicable listings from the approximately 160,000 WHOIS directory records will be made available on our server through WAIS. These records will then be corrected and updated by contacting (through e-mail) those listed and requesting correction/verification. Records will be transferred to X.500 as they are verified. Records which are not verified or which the listed individual doesn't want listed for any reason will be purged.

The InterNIC *Directory and Database Services* team (AT&T) has chosen X.500, because of its current availability, as the first implementation of the InterNIC distributed directory service. As other options (e.g., WHOIS++) become available, they will be evaluated and, depending on the outcome of that evaluation, will be implemented. The Directory and Database Services team will develop a user agent that supports the WHOIS protocol and that can search both the WAIS database and the X.500 Directory, thus providing a consistent user interface to the evolving directory.

As a general rule, the InterNIC will maintain no more than 50 entries that pertain to a particular organization in the X.500 *Directory Service Agents* (DSAs) that will be maintained on its servers. We will work to help organizations implement their own X.500 DSAs. With time, entries will moved from the WHOIS database to the InterNIC DSAs and the distributed DSAs. In parallel, new entries will be added to both the InterNIC DSAs and the distributed DSAs.

Documents

Related information is available via anonymous FTP on the InterNIC Directory and Database Services server `ds.internic.net` in the `/pub/internic-info` directory. The file `white-pages-position-paper.txt` contains further details and rationale, `x500desc.txt` contains a description of our X.500 services, and `organization-x500.template` contains the template that organizations need to complete for a listing in the InterNIC directory.

New service

The *InterNIC Directory and Database Services* has just fielded a WHOIS server which provides unified access to person data from a local database as well as from the MILNET person information held by the DISA NIC WHOIS server and the POC person information held by the RS WHOIS server. The server is available on all three DS machines. It listens on the standard WHOIS port and will query the DS, RS, and MILNET whois databases to provide a unified interface for queries about people. Users can access the DS WHOIS server using the "-h <hostname>" option of the *whois* command. For example:

```
whois -h ds.internic.net "Smith, John"
```

More information

Please direct all questions and comments to:

```
admin@ds.internic.net
```

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779